平成３年６月２日第３種郵便物認可　平成７年９月１日発行(毎月１回１日発行)　第７巻　第９号　通巻第64号

TRY COMPUTING

# トラ技コンピュータ

簡単にデータを収集し加工するためのノウハウ
特集 Windows時代のデータ計測と解析入門

新連載 Delphiプログラミング
新連載 理工系ユーザーのためのLinux入門

1995 9

「超高速」だけが再現できる、そんな感動に出会った。

マルチメディアの溢れんばかりの表現力を、超高速ペリフェラルがありのままに再現します。

3.5インチハードディスクユニット

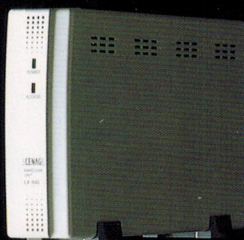
## SCENAGE LX series

最大容量1GBまでのすべてのモデルに平均シーク15mS以下、回転数4500rpmの高速ドライブを採用したSCENAGE LXシリーズ。PC-98シリーズ対応のDBモデルには、マルチベンダー機能をサポートしたSCSI-2インターフェイス「IF-2768」が標準添付され、抜群のコストパフォーマンスを発揮しています。さらにビギナーにも安心の使い易さと、MacintoshやDOS/Vパソコンにも対応した幅広いラインアップ構成で、すべてのパソコン環境をトータルにサポートします。

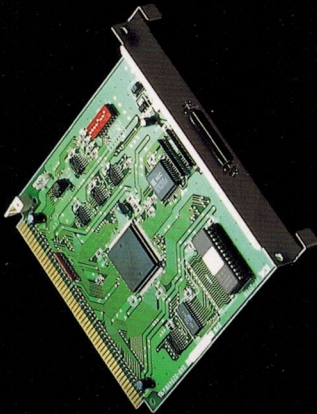
■メーカー希望小売価格

| | | | |
|---|---|---|---|
| LX-340-DB | ¥42,800 | LX-340-ISF | ¥52,800 |
| LX-340-M | ¥42,800 | LX-340-BLD | ¥37,800 |
| LX-540-DB | ¥44,800 | LX-540-ISF | ¥54,800 |
| LX-540-M | ¥44,800 | LX-540-BLD | ¥39,800 |
| LX-1000-DB | ¥79,800 | LX-1000-ISF | ¥89,800 |
| LX-1000-M | ¥79,800 | LX-1000-BLD | ¥74,800 |

SCENAGE LX series

MULTI VENDER

PC-9800シリーズ用 SCSI-2インターフェイスカード

## IF-2768

●データ転送方式として、DMA転送、FIFO転送、バスマスタ転送の3つの方式をサポート ●SCSI-2(Fast SCSI)を採用 ●SCSI-2対応の機器との間で、理論上最大10MB/秒のデータ転送が可能 ●SCSI-2はSCSI-1の上位規格なので、SCSI-1の機器も同時に接続可能 ●起動時のSCSIデバイスのサーチを高速化することにより、起動に要する時間を短縮 ●DMAチャネル、割り込みレベル、SCSI-IDの設定などがメニュー形式のソフトウェアによって簡単に設定可能 ●HDDパラメータ自動解析機能装備 ●他のインターフェイスカードに接続して使用していたハードディスクを再フォーマットすることなく使用可能(マルチベンダー機能)

IF-2768

4.4倍速CD-ROMユニット

## CD-660S

●転送レート:676KB/S
●アクセスタイム:150mS
●キャッシュメモリ:128KB

6倍速CD-ROMユニット

## CD-900S

●転送レート:922KB/S
●アクセスタイム:145mS
●キャッシュメモリ:256KB

※VideoCDの表示再生には専用のソフトウェア・ハードウェアが必要です。
※Photo CDの表示再生には専用のソフトウェアが必要です。

■メーカー希望小売価格

| | | | |
|---|---|---|---|
| CD-660S-N | ¥29,800 | CD-900S-N | ¥59,800 |
| CD-660S-NB | ¥39,800 | CD-900S-NB | ¥69,800 |
| CD-660S-M | ¥32,800 | CD-900S-M | ¥62,800 |
| CD-660S-ISF | ¥44,800 | CD-900S-ISF | ¥74,800 |
| CD-660S-BLD | ¥29,800 | CD-900S-BLD | ¥59,800 |

CD-900S

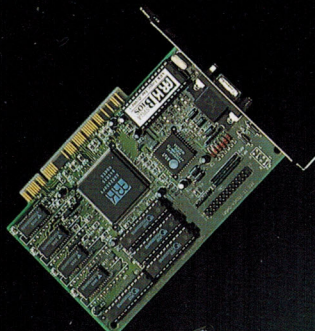
DOS/Vパソコン用PCIバス グラフィックアクセラレータ

## GP-2000PV-2M

●最大解像度1600×1200(16色、256色表示可能)
●64ビットARK2000PVを採用 ●PCIバスV2.0に準拠

GP-2000PV-2Mは、Windows上でのスクロール動作、動画再生もスムーズに行える先進的なアーキテクチャを採用。V RAMを採用せず、D RAM 2MBの搭載で高速描画を実現し、さらなるコストパフォーマンスを追求します。

■メーカー希望小売価格

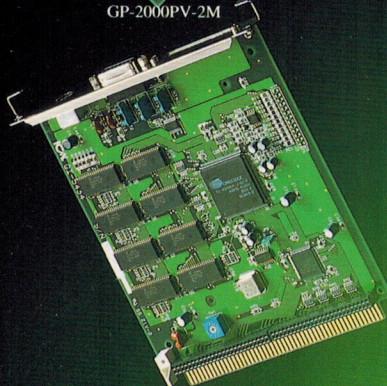
GP-2000PV-2M ………… ¥39,800

GP-2000PV-2M

PC-9800シリーズ用 グラフィックアクセラレータ

## GI-5434-2M/4M

●最大解像度1280×1024ドット ●フルカラー1,677万色表示
●最新64ビットのCL-GD5434(Cirrus Logic社製)を採用した2MBと4MBのD RAMを搭載した2モデル

■メーカー希望小売価格

GI-5434-2M (D RAM 2MB搭載モデル) ………… ¥24,800
GI-5434-4M (D RAM 4MB搭載モデル) ………… ¥32,800

GI-5434

※対応パソコン
●PC-98シリーズ:386以上、DOS 5.0以上、Windows3.1以上
●Macintoshシリーズ:68030以上、漢字Talk7以上
●DOS/V機:Future Domain社の推奨機、Windows3.1以上
※ラインアップ表記中の対応機種表現について
-DB=PC-98シリーズ用(IF-2768付き)
-N=PC-98シリーズ用(ICM製インターフェイスなし)
-NB=PC-98シリーズ用(ICM製インターフェイス付き)
-M=Macintosh用
-ISF=DOS/V用(ISAバスインターフェイス付き)
-BLD=DOS/V用(増設用)

INTELLIGENT INSTRUMENTATION™
知的生産性を高める新次元のデータ収集

# 計測・解析・制御システム

## 計測アプリケーションの開発時間を大幅短縮できます。

Visual DesignerはMESSCOMP'94における計測ソフトコンテストで優勝。
実力の程を、DASportでお確かめください。

**キャンペーン実施中！**

| | |
|---|---|
| ハードDASport | ￥139,000 |
| ソフトVisual Designer | ￥189,000 |
| | ~~￥328,000~~ |

**キャンペーン特価 ￥193,000**

ポータブル計測システム

## DASport

**机上から現場まで。データ収集が必要などんな場所でも使用可能。**

- ●100kHzアナログ入力
- ●チャンネル数：16、ゲイン：1、10、100
- ●トリガリング：アナログ／デジタル
- ●DI：8、DO：8ビット
- ●カウンタ：16ビット1個●DOS/Windows用ドライバ付
- ●寸法：203W×152.4D×63.5H(mm)

PCI-20450P-35　￥139,000

★アナログ出力、バッテリー付モデルもあり
★信号線接続オプション多種

計測・解析・制御・表示システム開発ツール

## Visual Designer

計測・解析・制御・表示用カスタムシステム開発の驚異的な時間短縮比を実現。オシロスコープ、PID、DSPラボ自動化、ライン自動比、モニター、ロガーなど用途は無限。GPIB、RS232通信、100MHzまでのA/D変換、フィルタリングなど多様な機能。Excel、1-2-3、LANによるリモート計測もサポート。

●3ステップでシステムを実行。
1.必要な機能ブロックを選択
2.ブロックを接続し、パラメータを設定
3.収集、データ処理、記録・表示・解析をクリック1つで実行

PCI-20901S-3　￥189,000

## 超カスタム、しかも今すぐ使えるポータブル計測システム

●Visual Designerの詳しい操作方法等を記載した日本語版ユーザーズマニュアルをご用意しています。
下記までお問い合わせ下さい。

**Visual Designerのセミナー開催日程**
**'95年8／25、9／8、9／22、10／6、20**

インテリジェント インストルメンテーション株式会社
〒222 神奈川県港北区新横浜1-3-10 新横浜IOビル7F
TEL.045-474-4162(代)　FAX.045-474-4167

A Burr-Brown Company

# ECN

ELECTRONICS & COMPUTER NEWS

9
1995

この欄に掲載されている情報は各企業からの広告メッセージです.
新製品、話題の製品、キャンペーン製品などをダイジェスト・スタイルでご案内しています.
ぜひご活用ください.
なお、掲載情報へのお問い合わせは、各々の綴じ込みリーダーズ・カードをご利用になるか、
直接広告主にご連絡ください.
また、ＢＯＸナンバのついた広告は、ＦＡＸ情報ＢＯＸで詳しい情報入手できます.
03-5570-1303にダイヤル後、メッセージに従って操作してください.



資料請求No.9001



資料請求No.9002



資料請求No.9003



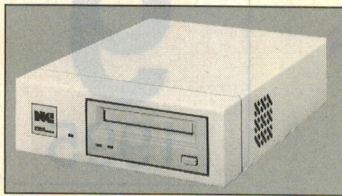
資料請求No.9004



資料請求No.9005

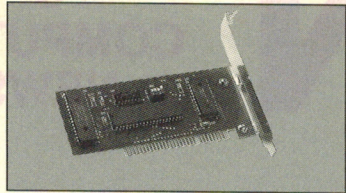


資料請求No.9006

## パソコン用ＤＡＴバックアップ N3200DATシリーズ

●パソコンのハードディスクを丸ごとバックアップ　（最大２ＧＢ）
●外付用サブシステムと内蔵型ドライブをラインナップ
●NetWare,WindowsNT等のネットワークＯＳにも完全対応
●148,000円（内蔵）178,000円（外付）

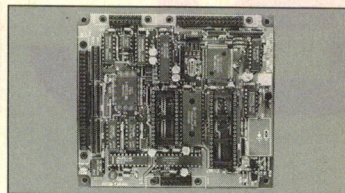
日商エレクトロニクス㈱PCプロダクト部
TEL 03-3544-8387　FAX 03-3544-8201
〒104　東京都中央区築地７－３－１

資料請求No.9007

## コンピュータウイルス防止カード VirusStop「バイラストップ」

◆拡張スロットに挿入するだけでウイルス侵入を防止 ◆メインメモリを占有しないため、メモリ不足や処理速度が遅くなる心配不要 ◆新ウイルスにも対応 ◆書込み禁止により書き換えおよび削除不可能 ◆DOSより前に監視開始 ◆定期的メンテナンスやアップグレード不要 ◆標準価格:¥30,000(税別)

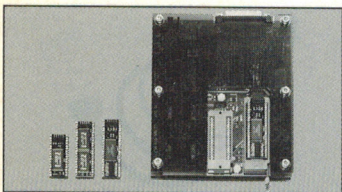
㈱ケンウッド 計測機器事業部 営業部
横浜市緑区白山1-16-2　TEL 045-939-7055

資料請求No.9008

## ワンボードコンピュータ ＭＹＫ80－ＣＡ64ＳＡ

64bitI/O＋4DAC＋4ADC　多機能ボード
●ＣＰＵ　ＴＭＰＺ84Ｃ015
●9.8304ＭＨｚ水晶発振
●32ＫＢＲＯＭ　32ＫＢＲＡＭ
●ＴＭＰ82Ｃ265　8bit×6ポート
●ＤＡＣ　ＭＡＸ506 8bit 4ch
●ＡＤＣ　ＭＡＸ150 8bit 4ch
●価格　¥40,000

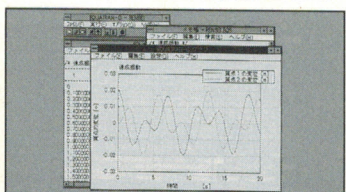
ミヤケ電子工業㈱　TEL 075-501-2022
〒607京都市山科区勧修寺福岡町280

資料請求No.9009

## 大容量マスクロムエミュレータ ＲＴ－System

<RT-system>は大容量マスクロムのエミュレーションを行なうためのエミュレーションチップと、そのライターです。
●ホストにはＤＯＳ／Ｖ機を使用
●EP-ROMよりも速い書込、消去時間
■RT-E8　32ピンDIP(8Mbit)
■RT-E16A　42ピンDIP(16Mbit)
■RT-E32A　42ピンDIP(32Mbit)

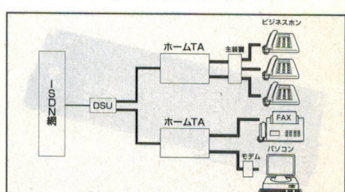
㈱ライズ 〒101 千代田区三崎町2-20-1
TEL 03-3238-1240　FAX 03-3238-7857

資料請求No.9010　FAXBOX No.121#

## 方程式解法ソフト EQUATRAN-GforWindows

ＥＱＵＡＴＲＡＮ(イコートラン)は数値計算をプログラミングせずに簡便に行うソフトで、いわば「技術計算のための簡易言語」です。線形・非線形連立方程式、常微分方程式を解くことができ、最適化問題や最小２乗法問題が扱えます。計算結果は、グラフ機能やレポート機能により加工が可能です。

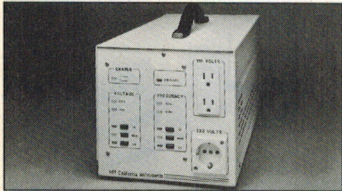
三井東圧化学㈱　TEL 03-3592-4190
〒100 東京都千代田区霞ケ関3-2-5

資料請求No.9011

## パソコン通信中に電話が使えます ホームＴＡ

◆電話中に同時にパソコン通信、FAX通信が可能 ◆電話回線１本で同時に２つの通話が可能 ◆ダイヤルイン機能あり
◆ワンタッチ高速リダイヤル機能付
◆手持ちの電話機、FAX、モデムが使用可
◆ＩＳＤＮ使用環境を低価格で実現
◆大きさ：タバコ２箱位
◆特別価格：¥50,000(税込み)

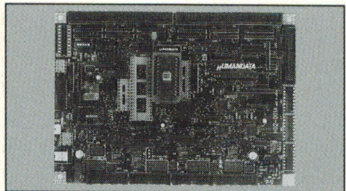
㈱エストマ　新宿区百人町2-24-13-103
TEL 03-5332-7531　FAX 03-5386-3490

資料請求No.9012

## 世界中の交流電源を１台で供給 1251WP型 ポータブル電源

●100Vコンセントに繋ぐだけで各国の電圧･周波数の得られる交流電源.●入力85～264V,47～63Hz.出力100/115/220/230/240V,50/60Hz.出力電力1250VA.●電圧･周波数確度1%,電源･負荷変動2%.●電圧±10%,周波数±3Hzを一発で設定できるテストモード.●ピーク容量大.●入力電流波形歪みはない.●重量13kg.

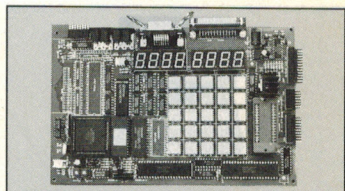
㈲セット ジャパン　TEL 03-3812-0240
〒113　東京都文京区本郷4-12-16-706

資料請求No.9013

## ROMICEに便利・ROM変換楽ちん! ソケットアダプタＸ

ソケットアダプタＸは、24～32ピンまでのROMソケットに差込み、ROMICEやROM交換をスピーディかつ安全に行うことのできるアダプタです。一度使ったら手離せない便利さです。付属ソケットの交換で24、28、32ピンが使用可能です。
本製品はFAXでのみ問合せに応じます。

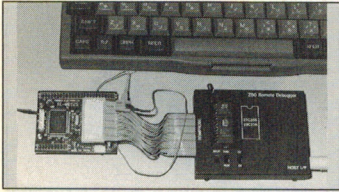
FAX 0726-20-2003　TEL 0726-20-2002
㈲ヒューマンデータ　〒567 茨木市
中穂積1-2-51　シャトー春日第3ビル2F

資料請求No.9014

## 作りながら学ぶマイコン入門基板 ＴＲＮジュニア

■豊富な機能：Z80系TMPZ84C013、ROM32K、RAM32K、25キー、8桁LED、セントロ、RS232C、A/D･D/A変換、スピーカ、ROMライタ、MIDI、120曲オルゴール、BASICおよびZビジョン対応。
■定価：基本キット12,000円、基本+拡張キット21,000円、完成品は+3000円

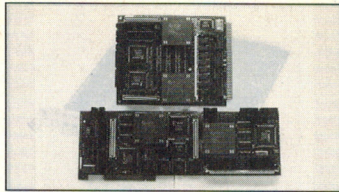
(株)ＥＳＰ企画 FAX 0774-65-1321(24時間)
TEL 0774-63-2743 (月～金AM10:00～12:00)
〒501-62岐阜県羽島市正木町三ッ柳123

資料請求No.9015

## Ｚ８０全機種対応！ ROMデバッガRM８０-DB

●簡単操作でＩＣＥライクな開発支援システム●RS-232Cを持たないＣＰＵボードのプログラムデバッグに使用可能●ＲＯＭ，ＲＡＭ全てのメモリ空間がプログラムデバッグに使用可能●セットアップが簡単でRESETやM1信号の補促不用●コントロールソフトを起動すると即プログラムデバッグ作業が行えます。

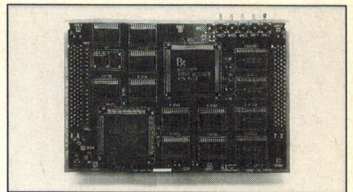
㈱専門教育研究所 TEL 03-5684-2906
〒113文京区本郷3-11-5 FAX 03-5684-2907

資料請求No.9016

## デュアルＤＳＰボード ＤＳＵ－Ｕシリーズ

DSU-Uシリーズは,PCI,98バスをサポートし、２個のTMS320C40を搭載したコストパフォーマンスの高いＤＳＰボードです。
●システムパフォーマンス ５５０HOPS
●D-RAM(32MB,2クロック), S-RAM(1MB)
●任意のユーザー回路の実装が可能
●目的に合せたモジュールが搭載可能
●ノーホストモード(ホストＣＰＵ不要)

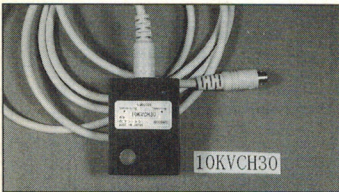
ハスク技研㈱ TEL 075-661-2662
〒601 京都市南区上鳥羽南唐戸町7

資料請求No.9017

## 目的に合せた拡張ができる ＤＳＵ－Ｕ専用モジュール

Ｖｉｄｅｏ－Ｇモジュール（写真）
●画像入力NTSC4ch●画像処理チップSALA搭載●画像処理空間max4096x4096
その他，下記モジュールがあります。
Ｐ－ＬＯＧモジュール●入力ch数１６ch
●サンプリングレートmax５００KHz
●出力CH数8ch
Ｓ－ＬＯＧモジュール●入力ch１６ch
●サンプリングレート max１０MHz

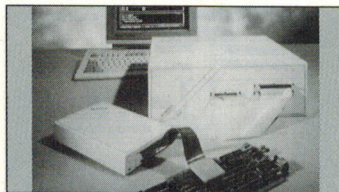
ハスク技研㈱（TEL・住所左記に同じ）

資料請求No.9018

## キーボードへのアタッチメント １０ＫＶＣＨ３０他

■10KVCH30［写真］¥4,500９月末迄モニタ値：DOSVキー配置交換が15ケ迄OSによらず可。不便なCTRL/ESCも好みの位置に■10KCV98VA ¥7,000好評発売中：手に慣れたPC98キーボードをDOSV OADG106キー（ミニDIN6P)へ中継■10KCVV98B ¥7,000（７月末予定)安価なDOSV用バーコードリーダが98用に。98Keyもつながるタイプ

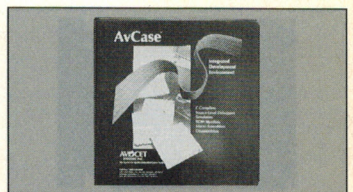
㈱テンショウ 〒583 羽曳野市白鳥3-4-3
TEL 0729-57-0789 FAX 0729-56-5159

資料請求No.9019

## ボーダレスで使用可能なＩＣＥ Pentica ＭＩＭＥシリーズ

●英国Pentica社のインサーキットエミュレータは信頼性・新チップ対応の速さ等で欧米マイコン開発者の標準機として定着●Windows版Ｃソースコードデバッガ●トリガポイント、イベント等のパフォーマンス分析ツールや計測インスツルメンツ提供●Motorola系及びHitachi系のＭＰＵには特に強く各種チップ に対応

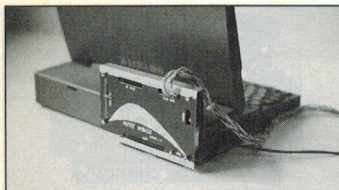
㈱ソーテック工人舎(事) TEL 045-661-7360
〒231 横浜市中区太田町４－５５

資料請求No.9020

## AVOCETマイコン開発ツール ＡＶＣＡＳＥⅡ６８ＸＸＸ

従来のAVCase68Kのアッパーバージョン●68000/10/20/30/40/,,683XX,ＣＰＵ32までフルサポートするＣコンパイラ、アセンブラ、シュミレータ／デバッガを含む統合開発ツール●ＩＥＥＥ６９５支援拡張メモリ、コプロセッサ対応●コンパイラやアセンブラのみの販売も可能
68Ｋ以外のターゲット各種ラインアップ

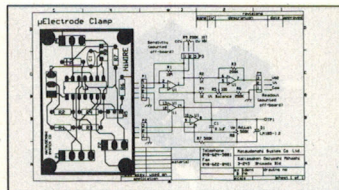
㈱ソーテック工人舎(事) TEL 045-661-7360
〒231 横浜市中区太田町４－５５

資料請求No.9021

## ノートPC用計測＆制御モジュール ＮＯＴＥ－ＷＩＮＧＳ

◆ノートPC98（拡張バス110P）用の小型マルチI/O。Lotus123等と組合わせて、計測処理用のＩ／Ｏサンプルソフト付
◆A/D.12bit×8ch,D/A.12bit×2ch DiO.32bit,CT.2ch,EXP-I/OバスＣＮ付
◆拡張バス直接固定型NTML-98D ¥43000
◆オプション．．データロガーソフト他

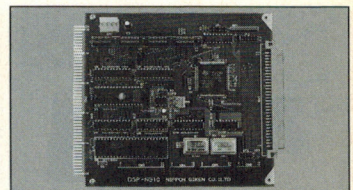
㈲ローム理研
〒431-04 静岡県湖西市新所原2-6-7
TEL 053-572-3961 FAX 053-577-1969

資料請求No.9022

## 回路・アートワーク ＣＡＤ ＨｉＷＩＲＥ Ⅱ

BBS：048-622-8407,N81,9600,開局
DemoSoftDown,ゲスト歓迎
ID取得者Schoo l VersionDown 可
UpDATAから低価格基板製作受託
1)HiWIREⅡEDT ９万
2)Autorouter for Dos １２万
3)Autorouter for WINDOWS １６万
割引 1)＋2) 20万円 1)＋3) 23万

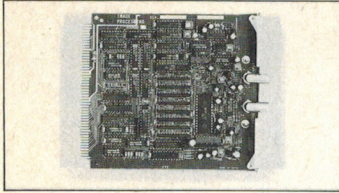
松電子システム㈲ TEL 048-624-5881
〒331埼玉県大宮市三橋3-245島田ビル1F

資料請求No.9023

## デジタル信号処理のパートナー 超高速ＤＳＰボード

長年の実績が貴方のご要望にお応えします。波形解析や適応制御等々、必要な項目・内容を提示いただければ、ハード及びソフトの設計や製作をして提供いたします。ワンボードからシステムまで、簡単なものから複雑なものまで、初歩から専門段階まで種々サポートできます。

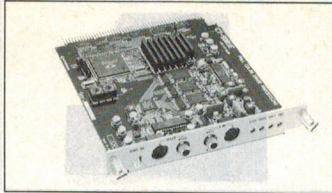
㈱日本技研
TEL 0182-32-0506 FAX 0182-33-6533
〒013 秋田県横手市赤坂字大道添3-2

資料請求No.9024

## フレーム・メモリ(モノクローム)
## CT-9800B

特長●1画面の記憶と表示出力：512(H)×512(V)×8bit 1画面 or 512(H)×256(V)×8bit 2画面●入出力は各1ケのBNCコネクタ●入力画面への計測用スケールや文字・図形のスーパーインポーズ●ダイレクトメモリマッピングによる高速で簡単なアクセス●対応機種：PC98, PC486, FC98●サンプルソフト付●オプション有

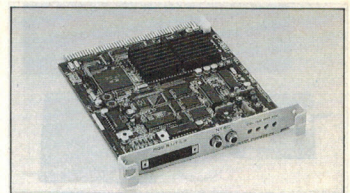
サイバーテック(株) TEL 06-372-5558
〒531 大阪市北区本庄東3-9-15

資料請求No.9025

## フルカラー・フレーム・メモリ
## CT－9801E

●シンプルデザイン＆高品質●フルカラー1677万色●XY座標で使い易い512(H)×512(V)画素を採用●入出力はNTSCコンポジット、S端子信号●アスペクト比1：1、4：3切換機能●1/2縮小機能●ダイレクトメモリーマッピング●CT-9801Cプログラムコンパチブル●PC-9801全機種対応●低価格

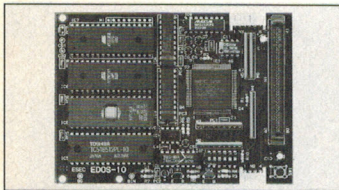
サイバーテック(株) TEL 06-372-5558
〒531 大阪市北区本庄東3-9-15

資料請求No.9026

## フルカラー・フレーム・メモリ
## CT-9801F(新製品)

特長●1677万色フルカラー●XY座標で使い易い512(H)×512(V)画素で2画面、又は640(H)×512(V)画素で1画面●入出力：NTSC、S端子、RGB信号、PC画面へのスーパーインポーズ●ダイレクトマッピング(2Mバイトのプロテクトメモリー又は128KB×16のバンクメモリー、32ビットパックドピクセル)●PC98全機種対応

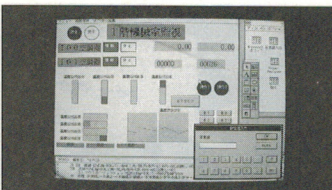
サイバーテック(株) TEL 06-372-5558
〒531 大阪市北区本庄東3-9-15

資料請求No.9027

## ARCNETサポート
## 組込みカードパソコン

●アイス、ライタ不要　86×118㎜
●FAに最適なARCNET-LANをサポート。パソコン1台で、各ノードのソフト開発を可能とするLANスコープを用意。
●ROM-DOS(データ・ライト社製)搭載。開発キット用ソフトE-TERM有。
●組み込み制御用LANに最適。

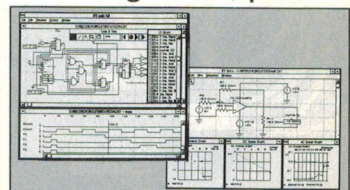
日本エセック㈱
TEL 03-5256-3166 FAX 03-5256-3167
〒101 千代田区外神田2-10-8 富士ビル

資料請求No.9028

## PC用制御グラフィックソフト
## Windows版“カンタ”

■グラフィック画面で作画。テキストエディタでも作画可能。■最大50画面、1画面100要素。■I/O制御はシーケンサとリンク可。■シーケンス演算式が記述可能。
■ネットワークで装置間を連結可能。
■98、DOS/V動作可能ソフト有。
■FAパソコン、ボードなど各種多数を取り扱い。

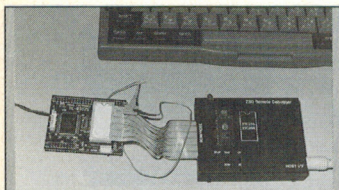
㈱アイケーシステム TEL FAX 0568-75-8886
〒485 愛知県小牧市小木東2-98

資料請求No.9029

## ﾃﾞｨｼﾞﾀﾙ/ｱﾅﾛｸﾞ回路シミュレータ
## B²Logic／B²Spice

個人の研究者や技術者むけに開発されたシミュレータで研究者自身が個人で購入できるように適正な価格になっています。従来のこの種のソフトウェアに比べ大変安価ですが、設計できる回路の大きさやノード数など機能上の制限はありません。
B²Logic ¥39,800／B²Spice ¥29,800
●トラ技広告頁もご覧下さい●

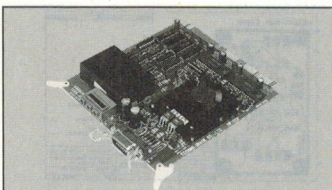
(有)そらコンピュータ・プロダクツ
TEL 045-262-1097 FAX 045-232-1637

資料請求No.9030

## 多機能高速Z80リモートデバッガ
## ZDB101

Z80全機種に対応可能なROM型デバッガ東芝、ザイログ、日立、川鉄、10M-NoWAITOKシリアルポートが空いてない機種でも可SystemLoad社ZVisionRemote(MINI可)または専門教育研究所RM80にも対応可能ROM空間へのユーザープログラムロードICEのような使い勝手でROM化可能なプログラムのままデバッグ　¥35,000

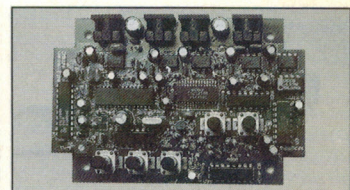
㈱ホロソンテック TEL 03-3288-9951
〒102 千代田区九段北1-12-3九段北ビル

資料請求No.9031

## PC98用プログラマブル電源ボード
## 『OP12CP』シリーズ

●回路動作試験、FA/LA検査器用途に最適
●分解能：128bit ●出力数：1 or 2
●出力電圧：±10,±5,±2.5,+10,+5V選択
●出力総合電流:+1A,-750,±450mAMax
●電流シンク機能、フェール検出機能、トリガ出力機能、出力保護機能
■～¥55,000■取扱店募集中

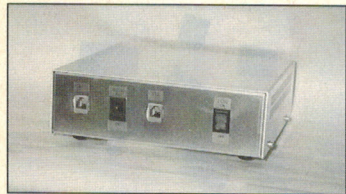
テクノ・グレース㈲
〒917 福井県小浜市湊11-15-3
TEL 0770-52-7724 FAX 0770-52-7720

資料請求No.9032

## 映像信号の加工に！
## MOVE－PICT

◆NTSC表示位置(HV)移動、輝度、色調整基板(MOVE-PICT Ver1.1)……¥15,800
★位置:±50% ★輝度:0～100%
●Y/C入力→コンポーネント(R-Y,B-Y,Y,SYNC)変換基板……………¥8,800
●速度カウンタ基板(制限付)……¥13,800
●PWM(FET)ドライブ基板(DC20A)·¥5,800

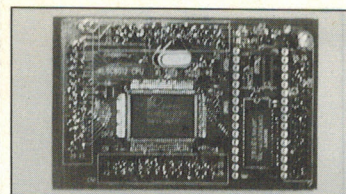
(有)ログファーム
TEL 08847-7-0099 FAX 08847-7-2118
〒779-23 徳島県海部郡日和佐町245-27

資料請求No.9033

## NTTダイヤルインサービス端末用 電話交換シミュレータ

■ダイヤルインサービス端末を対象に相互接続および通信テストができます。
●DP(20PPS)･PBを自動識別 ●スイッチ切替でアナログ通信端末の試験も可能
●同機能内蔵FAX装置などの販促用途や開発ツールに最適 ◎NTT回線接続不可
●標準価格 ¥148,000(納期約1週間)

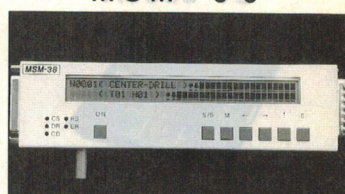
㈱アップロード
〒734 広島市南区西旭町14-25
TEL 082-254-8400 FAX 082-254-8777

資料請求No.9043

## Z80コンパチブルCPUボードキット SK－8012シリーズ

●川崎製鉄製KL5C8012を使用 ●同クロックZ80の約4倍の処理能力 ●ボード上にRAM、ROM各最大1Mビット搭載可能
●CPUクロック：9.8MHz ●I/Oポート：40ビット ●寸法：80×53mm
■AX(外付RAM＋開発ツール付) ¥9,800
■BX(外付RAM＋資料付) ¥5,000
■CX(内蔵RAMのみ) ¥3,500

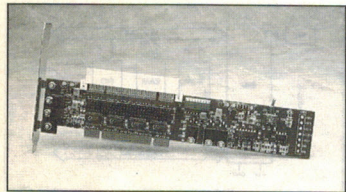
㈲システムロード TEL 048-475-0367
〒352 埼玉県新座市東北2-29-11

資料請求No.9044

## 超小型プロトコルモニター MSM－38

I/F：RS232C、RS422A、RS485
速度：非同期 300～19200BPS
表示：JIS、ASCII、HEX ISO、EIA(NC)
40×2行、記憶2560B(P･OFF保持)
電源：電池/AC100V オートパワーオフ
出力：保持データ出力
■本体価格 38,000円

マイクロサポート(株) TEL 06-399-5577
〒532 大阪市淀川区宮原1-19-10

資料請求No.9045

## ボード開発，検査，保守の必需品 PCI,ISA,VMEエクステンダ

◎アイソレーションタイプ（端子付）
遅延250psの最新ICを用い，しかもPCの電源をOFFにせずにボード着脱可。
・PCI532，332(写真)・PCIX32－5,－3
・ISAIX16 ・VLBIX ・VME100
・64bitPCI ・PMC
◎ノーマルタイプ（プローブ端子付）
・PCIEX4－5H－D,－3H－D

立野電脳(株) FAX 0428-77-7010
E-mail：GGC00374@niftyserve.or.jp

資料請求No.9046

## 小型で低価格の普及型電力計 デジタル電力計

●コンセントに差し込むだけで面倒な配線が不要
●小型・軽量で取扱いが簡単
●ホールセンサー直接乗算方式で真の電力実行値表示
●その他デジタル電力計ファミリーもあります ●価格￥２５，６００～

(株)ユー・アール・ディー
TEL 045-502-3111 FAX 045-502-3632
〒230 横浜市鶴見区本町通4-169-3

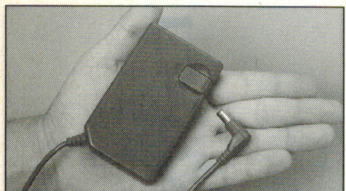
資料請求No.9047

## 不正コピー防止ツール 持出厳禁 V1.5

･コピープロテクトツール･Windows DLL呼出･DOSのCOM,EXEに直接プロテクト･強力キーフロッピ･手軽な運用キーハードディスク･ランタイムフリー･納品，販売，見本ソフトに最適･企業，学校，内部ソフトに使用可。 ■価格各48,000円
☆ソフト開発承ります。データベース、DOS,Windows,Z80,86,ROM化

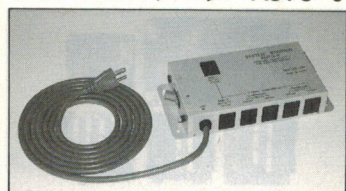
㈱ポトス TEL 078-753-6443
〒655 神戸市垂水区塩屋町4-1-7-303

資料請求No.9048

## 小型で携帯しやすいACアダプタ PSA-10・PSA20

小型の平面トランスと小型制御ICを採用し、スリムで軽量なスイッチングACアダプタです。ACプラグは写真の様に筐体に収納できます。PSA10は出力が5～15Vまで、入力100Vの10W。 PSA20は入力220Vまで20W仕様。ノートパソコンに最適。

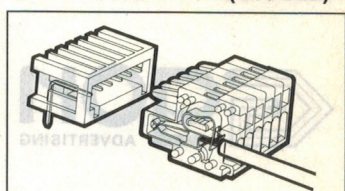
三真物産(株)
TEL 06-229-0302 FAX 06-229-0339
〒541 大阪市中央区瓦町4-3-910

資料請求No.9049

## パソコン通信革命 システムスターターAUTO-6

■電話の呼び出しベルが鳴ると自動的に出力コンセントがオンしてシステムを立上げ無人運転を開始、通信終了後は自動オフする。終了後も通信があったことをアクセスランプで表示。
■その他の機種として、CPUの電力消費を検出して周辺装置の電源を自動オン／オフするタイプとノートパソコン専用。

(株)エヌ・イー・エス TEL 0286-82-8841
〒329-13栃木県塩谷郡氏家町氏家3488-4

資料請求No.9050

## MULTI CONNECTION SYSTEM 733シリーズ(新製品)

従来の231シリーズに、極間ピッチサイズが半分の2.5mmピッチタイプが新登場し、さらに高密度実装を可能にした。
●定格電圧：250V
●定格電流：４Ａ
●使用電線範囲：0.08～0.5㎟
●極間ピッチ：2.5mm

ワゴジャパン㈱
TEL 0423-30-7511 FAX 0423-30-7575
〒183 東京都府中市日鋼町1-1 Jタワー

資料請求No.9051

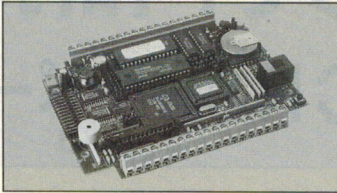
## DSP専門パワーシステムハウス ディジタル信号処理

# DSP
ソフト＆ハード

ＤＳＰソフト開発からハード開発・製造まで一貫して受注できます。ＤＳＰソフト開発単独のご依頼も歓迎いたします。
●音声処理＝分析・合成・認識・識別・ボコーダ・ＶＯＸ●音響処理＝雑音除去・ＡＮＣ・信号検出●伝送処理＝エコー消去・モデム・セキュリティ・誤り制御●画像処理●自動制御●計測●解析 etc

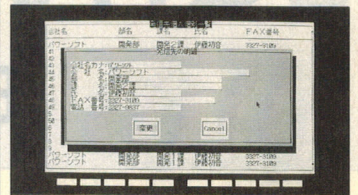
㈱スペクトラ TEL 0272-63-1178
〒371 群馬県前橋市広瀬町1-29-17

資料請求No.9052

## メカ制御からデータ収集まで！ C言語コントローラ

Ｃ言語コントローラは、シーケンサに代わりメカ制御ができ、アナログ制御や計測器からのデータ収集など組込み用途に応じて豊富な種類があります。
開発はWindows上で編集からデバッグまで快適にでき、ＩＣＥを必要としないのでとても低価格です。

カナース・データー㈱ SH事業部 〒101
千代田区神田佐久間町2-11-1笛木ビル3F
TEL 03-3851-2680 FAX 03-3851-4568

資料請求No.9053

## 簡単・低価格・Ｃ言語用ライブラリ Power Window

●簡単で低価格なＣ言語（MS-C,BCC）用の画面制御ライブラリです。
●同一のEXEがDOS/Vと98で動作します。
●約200個の使いやすく有用な関数を用意しています。
●導入日よりプログラムの開発工数を一気に短縮できます。
●価格 ¥29,800（税別） マニュアル付

(株)パワーソフト世田谷区大原2-25-4-103
〒156 TEL 3327-3109 FAX 3327-9837

資料請求No.9054

## コンピュータベース計測製品満載 最新日本語版総合カタログ

パソコン主体の計測制御及びFAシステムを構築するために必要なハードウェア、ソフトウェアを選択する際に役立つ新カタログ。NEC PC-98やDOS/V機に対応する新製品が満載。GPIB、データ集録ボード、計測制御用ソフトLabVIEW、LabWindows/CVI、数値解析ソフトHiQなど。

日本ナショナルインスツルメンツ㈱
〒105 東京都港区芝公園2-4-1 秀和芝パークビルB館-5F TEL 03-5472-2970

資料請求No.9055

## H8/500シリーズマルチタスクＯＳ ROMOS

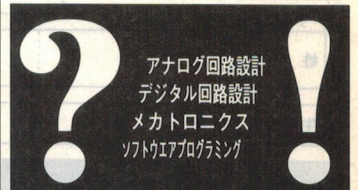
OSの提供する機能と標準システムコール数(カッコ内)

| | |
|---|---|
| ◎タスク管理機能 | メールボックス機能 (5) |
| タスク管理機能 (13) | ランデブ機能 (4) |
| タスク付属同期機能 (9) | ◎割り込み管理機能 (3) |
| ◎同期／通信機能 | ◎例外管理機能 (3) |
| イベントフラグ機能 (6) | ◎メモリプール管理機能 (4) |
| セマフォ機能 (6) | ◎拡張SVCサポート機能 |

●μＩＴＲＯＮ仕様に準拠
●容易なシステム構築
ＯＳの機能モジュールはライブラリ形式で提供
●高速なリアルタイムＯＳ
１０MHz動作時、割込みマスク時間１５μs、タスク起床時間４４μs
■ 価 格 26,000円～

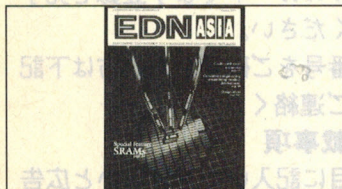
(有)システムワン TEL 0776-38-7290
〒910-37 福井県福井市浅水二日町403

資料請求No.9056

## 企画・開発・設計・試作 総合エンジニアリング

設計／開発は外注できます
アイディア／企画の具体化
■AV製品の総合設計
■無線通信機器の総合設計
■計測ロボットの開発
■メカトロニクス機器
■特注計測器／制御機器

(株)ケント電子設計
TEL 03-3495-0861 FAX 03-3495-0493
〒153 目黒区目黒2-11-11 HKビル3F

資料請求No.9057

## アジア向け電子技術専門誌 EDN ASIA

EDN ASIAは、米国最大のカーナーズ出版社が発行するエレクトロニクス技術情報月刊誌です．アジアの厳選した技術者28,000人に最先端の技術・製品情報・市場ニュースを提供します．貴社のアジア地域向け広告媒体として，ぜひ当誌をご利用ください．お問い合わせは下記へ

㈱ダイナコインターナショナル
〒165 東京都中野区沼袋2-1-7-101
TEL 03-3389-1822 FAX 03-3389-1761

資料請求No.9058

## 中国エレクトロニクス技術情報誌 EDN China

EBA・EDN Asiaに続いて、昨年の９月に創刊されたのがEDN Chinaです。中国本土だけをターゲットとしたこの媒体は、中国の主要都市に約20,000部配布されています。主な読者層は、エンジニアマネジメント及び、強い決定権を持つ工場長クラスを中心にしています。

㈱ダイナコインターナショナル
〒165 東京都中野区沼袋2-1-7-101
TEL 03-3389-1822 FAX 03-3389-1761

資料請求No.9059

## 韓国エレクトロニクス技術誌 『電子技術』

本誌は、ＣＱ出版社発行の「トランジスタ技術」と編集提携している韓国有数のエレクトロニクス技術誌です。いまや日本とともに世界の半導体市場で大きなシェアを占める韓国のエレクトロニクス分野で多くの読者を獲得しています。本誌のお問い合わせは下記まで。

㈱ダイナコインターナショナル
〒165 東京都中野区沼袋2-1-7-101
TEL 03-3389-1822 FAX 03-3389-1761

資料請求No.9060

通常請求用

# CQFAX資料請求システム

## ECN専用ページ

「CQ(Convenient＆Quick) FAX資料請求システム」は、お忙しい読者にとり頼もしい情報収集方法です。手順は簡単。まず、このページをコピーして 空欄の5項目にご記入ください。あとはFAXで小誌まで送信するだけです。
いつでも何度でも、すぐ請求できる「CQFAX資料請求システム」をどうぞご活用ください。

**資料請求カード　ECN　1995年9月号**

当社記入欄
1 VOL 0372
2 受付
3 登録内容変更箇所No.

カード有効期限：1995年9月末日迄
送付先は勤務先に限ります。

4 請求状況 2 （1＝新規　2＝登録済　3＝登録内容の変更）

5 会員番号 CQ
※会員の方のみご記入してください。

6 フリガナ
7 姓名　姓　名
8 生年月日 19　年　月　日
9 〒　都道府県　区市郡　区町村
10 勤務先住所
11 会社名
12 部署名
13 勤務先電話番号　市外局番　局番　電話番号
14 役職
15 所属部門
16 担当分野
17 業種
18 従業員数

19 広告資料請求欄　請求No.の欄に資料請求番号を記入し、A.B.C.のご希望の項目に○をつけてください

| 希望項目＼請求No. | 90 | 90 | 90 | 90 | 90 | 90 | 90 | 90 | 90 | 90 |
|---|---|---|---|---|---|---|---|---|---|---|
| 購入希望 | A | A | A | A | A | A | A | A | A | A |
| 説明希望 | B | B | B | B | B | B | B | B | B | B |
| 資料希望 | C | C | C | C | C | C | C | C | C | C |

| 希望項目＼請求No. | 90 | 90 | 90 | 90 | 90 | 90 | 90 | 90 | 90 | 90 |
|---|---|---|---|---|---|---|---|---|---|---|
| 購入希望 | A | A | A | A | A | A | A | A | A | A |
| 説明希望 | B | B | B | B | B | B | B | B | B | B |
| 資料希望 | C | C | C | C | C | C | C | C | C | C |

ECN専用

**ご利用の手順**

①資料請求会員番号の確認
②このページをコピーする
③空欄の5項目に記入する
1.会員番号
2.ご氏名
3.生年月日
4.資料請求番号
5.請求希望項目
④03-5395-2104に送信する

**※利用資格**
資料請求会員の方だけがご利用になれます。会員未登録の方は、綴じ込みのカードでまず登録を完了してください。
会員番号をご確認したい方は下記までご連絡ください。

**※記載事項**
5項目に記入いただけないと広告主に連絡できません。

# FAX番号：03-5395-2104

●お問い合せは下記まで●
CQ出版株式会社 広告部広告営業課 資料請求係　TEL.03-5395-2132

No.64

# トラ技コンピュータ CONTENTS 1995年9月号

簡単にデータを収集し加工するためのノウハウ

## 特集 Windows時代のデータ計測と解析入門



## 一般＆連載



## DOS/V＆Windowsのページ

# TRY COMPUTING

## トレンド



## インフォメーション



## Readers' SQUARE



## 特集について

Windowsのバージョンも上がり，OS自体もパワフルになってきました．従来ワークステーションでなくてはできなかったデータ解析もパソコンで可能になってきました．高機能なデータ収集ボードも低価格で入手できるようになっています．これらを活かすためにはアイデアがたいせつです．今回の特集では頁のゆるす限りたくさんの事例を解説しましたが，パソコンによる計測システムの構築の参考になれば幸いです．

## 次号のお知らせ　9月18日発売

**特集　Visual C++ MFCの研究**

Windowsの開発環境は，Visual C++とVisual Basicが定番ですが，Visual Basicに比べてVisual C++はアマチュアには手を出しにくいものです．特にMFCと呼ばれるクラスライブラリは，どのように使えばよいのかなかなか理解できません．そこで，次号ではグラフィックに関するクラスライブラリに焦点を当てて，MFCを使いこなすノウハウをやさしく解説します．ご期待ください．

## 編集部から

＊地方の時代といわれて久しいが，都市に集中する人口や機能はなかなか地方に分散しそうもない．いくら日本が狭いといっても，分散すればもっと安く広い家に皆が住めるはずなのに．集まっていないと安心できないのが日本人なのだろうか．これだけインターネットが騒がれているのは，本当ならサテライトオフィスなどを増やすことが主であってもよいはずだが，どうもそうではないらしい．〈山〉

＊4月1日の情報だったのでジョークだと思っていたのですが，ほんとうにSolaris 2.4のCD-ROMが先日自宅に送られてきました．昨年の末に雑誌の付録についていたDaytona-CD-ROMを3枚をSUNに送るとSolaris 2.4(SPARC版)が無料で，もしくは2万円でPC版が入手できるという話だったのです．IPCだってちゃんと動いている．うれしい！ HotJavaを動かそう！ 投資したのは850Mバイトのハードディスクだけ．〈ヨ〉

＊首都機能の移転が計画されています．なるべく地震の少ない所というのが，重要なポイントの一つになるでしょう．それと，できることなら，最近話題になっているネットワーク/コンピュータ・アーキテクチャのように，省庁の「分散処理化」も考えてほしいと思います．農林省は山形県に，大蔵省は大阪府に，文部省は長野県になど，いいと思いませんか．〈Ⓒ〉

＊友達のほとんどが仕事に対して夢を持っている．その中で現在就職活動をしている人がいる．親の手前で受けた会社に内定をもらったが，目指していることとは違いすぎて断ったそうだ．夢を掴める人は一握りだが，この不況の中でも妥協できなかった彼等を素晴らしいと思った．私も夢はたくさん持っているけれど，思っているだけで何の行動もしていない．時の流れの速さに焦ってばかりでなく努力しなければいけないな．〈中〉

**トラ技コンピュータ　1995年9月号**

発行所　CQ出版株式会社
〒170　東京都豊島区巣鴨1-14-2
☎03-5395-2124(編集)/03-5395-2141(営業)/03-5395-2132(広告)
発行人　蒲生良治　　編集人　山岸誠仁

1995年9月1日発行
振替　00100-7-10665
定価は裏表紙に表示してあります

印刷/製本 三共グラフィック㈱　表紙デザイン ㈱アイドマ・スタジオ　トビラ・カット MINO MIED　イラスト 神崎真理子

TRY COMPUTING トラ技コンピュータ

# SPECIAL

1995年9月号

簡単にデータを収集し加工するためのノウハウ

# 特集　Windows時代のデータ計測と解析入門

データ収集という作業は，コンピュータのもっとも得意とする分野です．したがって，MS-DOSが全盛の時代から，データ収集用の拡張ボードとして，A-Dコンバータボードやディジタル入出力ボードなどがたくさん販売されてきました．そして，ユーザーはBasicやC言語を用いたプログラムによってデータを集め，それを整理して利用してきました．

ところが，最近ではWindowsを利用できるようになってきたため，単にデータを集めるだけでなく，データの解析までをパソコンで簡単に行うことができるようになりました．

すなわち，PC98かDOS/Vマシンかといったハードウェアを問わず，全く同じ処理を行うことができるので，たいへん効率のよいシステムを組むことができます．

そこで本特集では，いろいろな用途にあった専用のデータ収集ソフトの活用法に加えて，Excelのように日常的に使われているソフトを利用した活用事例を整理して解説しました．

コンピュータにあまり詳しくない読者が初めてシステムを組んで実験を行い，そしてデータを収集して，グラフを描いたりFFT解析をすることが簡単にできるように構成を考えました．

## CONTENTS

## 特集 目次

# 第1章

## これからのパソコンによる計測システムの環境を考える

# Windows環境における計測，制御，解析とは

臼田昭司

昨今，パソコンによる「計測・制御・解析」環境(以下,計測環境)も一昔前と比べるとずいぶん変わってきました．ハード面ではPC98マシンが中心の時代からDOS/Vマシンへの移行，ソフト面ではなんといってもDOS環境に加えてWindows環境の拡充があげられます．

現在市販されているDOS/Vマシンのほとんどは Windowsがプリインストールされており，これにより**図1**に示すように計測環境が大きく変わりつつあり，環境さえ整えば比較的スムースに仕事にとりかかることができます．

### ● フレキシブルでパワフルなツールを活用する

これまでは，目的のソフトウェアの実行環境を得るために，自分でBASICやC言語などを使ったDOSレベルでのプログラミング作業が必要でした．

しかし，Windows環境では各種専用のツールがハードウェアメーカーからも提供されるようになり，プログラミングの知識がそれほど必要でなくなってきたので，データ計測もよりポピュラになってきました．

また，これらツールには解析機能も内蔵されており，解析結果をビジュアルに見ることができます．そのため，BASICやCなどの言語を使いこなせなくても，目的とする計測環境が比較的簡単に実現できるようになりました．

この種のツールは，ハードウェア(A-Dコンバータ，GPIBなど)と一体で使用することが前提になっていますが，機能の充実によりこれまで保有してきたデータ(例えば，ASCIIデータなど)を取り込むことが可能です．あらためてデータの解析をしなおしたりすることなどができます．

また，解析ソフトによっては，複数のメーカーのハードウェアをサポートしていたりします．このように，これまでの資産を有効活用することができます．

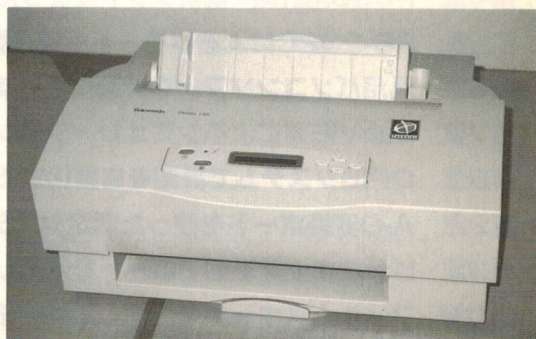
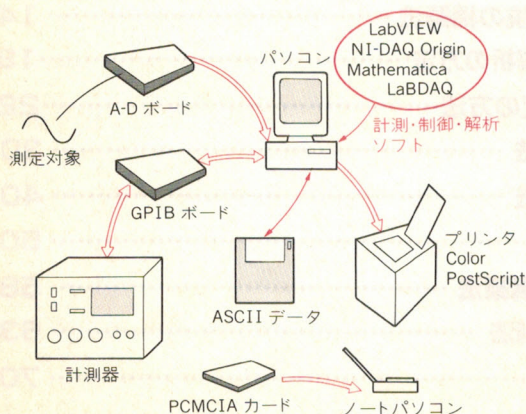
### ● ノートパソコンも活用

ハードウェアも従来のパソコン本体のバス拡張スロットに装着するボードタイプに加え，ノートパソコン対応のPCMCIA仕様のカードタイプが新たに加わりました．目的によっては，ノートパソコンを使えばよりポータブルに計測・制御・解析ができるというわけです．

ノートパソコンといっても，最近はCPUに486DX2 (66 MHz)やPentiumプロセッサを搭載し，ハードディスクも500 Mバイトクラスになってきており，これらは計測環境として充分なものとなっています．

### ● プリンタのカラー化

また，計測環境のインターフェースの一つであるプリンタはモノクロ出力が中心でした．しかし，最近になってカラー化が見直されてきました．

〈図1〉「計測・制御・解析」環境の例

〈写真1〉カラーPostScriptプリンタ PhaserI40

〈表1〉▶
インテルOverDriveプロセッサ製品リストの一部(パソコンの機種と使用できるOverDriveのリストも入手できる)

● IntelDX2 OverDrive プロセッサ

| 製品名 | 対応CPU | 希望小売価格(税抜) |
|---|---|---|
| JBOXDX2ODP50 | i486SX 16MHz, i486SX 20MHz, i486SX 25MHz i486DX 25MHz | 17,800円 |
| JBOXDX2ODP66 | i486SX 16MHz, i486SX 20MHz, i486SX 25MHz, i486SX 33MHz i486DX 25MHz, i486DX 33MHz | 23,800円 |
| JBOXDX2ODPR50*1 | i486DX 25MHz | 17,800円 |
| JBOXDX2ODPR66*1 | i486DX 25MHz, i486DX 33MHz | 23,800円 |

(*1) この製品はパソコン本体のCPUと差し替えるタイプのOverDriveプロセッサ

● IntelDX4 OverDrive プロセッサ

対象のODP　元のCPU

| 製品名 | 対応CPU | 希望小売価格(税抜) |
|---|---|---|
| JBOXDX4ODP75 | i486SX 16MHz, i486SX 20MHz, i486SX 25MHz i486DX 25MHz | 24,800円 |
| JBOXDX4ODP100 | i486SX 16MHz, i486SX 20MHz, i486SX 25MHz, i486SX 33MHz i486DX 25MHz, i486DX 33MHz | 36,800円 |
| JBOXDX4ODPR75*1 | i486DX 25MHz | 24,800円 |
| JBOXDX4ODPR100*1 | i486DX 25MHz, i486DX 33MHz | 36,800円 |

(*1) この製品はパソコン本体のCPUと差し替えるタイプのOverDriveプロセッサ

〈写真2〉ODP(製品名JBOXDX4ODP75)本体

モニタ画面だけではなく，プリントアウトもカラーによりビジュアルに表現することが必要になってきました．数種類の計測データなどはカラーにより識別することによって大変見やすくなります．また，プレゼンテーションのときなどは説得力のある情報伝達手段になります．

Windowsで使えるカラープリンタも低価格になってきました．また，PostScriptプリンタも市販されるようになってきています．昨今，PostScriptプリンタはずいぶん安価になり，コストパフォーマンスが向上しました．カラーのPostScriptプリンタも同様です(**写真1**)．これからは計測データの印刷もカラー化になっていくことが期待されます．

特集では，このように計測環境が変わっていく中で，筆者らが体験・試用した事例を中心に，「Windows環境を中心とした計測・制御・解析」について紹介します．ここでの紹介は一部にすぎませんが，できるだけ広い範囲にわたってとりあげました．ほかにもいろいろなバリエーション(ソフト&ハードの種類)があり，アプリケーションもたくさんあります．

このようなことから，計測環境はこれからも“より使いやすく”をコンセプトにますます変貌していくことでしょう．

## CPUのパワーアップで計測環境を向上させる

計測制御用のCAE(Computer Aided Engineering)ツールを使う環境として，CPU自体の高速演算能力が望まれます．加えて，数値演算コプロセッサ(以下，コプロセッサ)を備えていることが推奨されます．これについては，たとえば，National Instruments社の計測制御ツールであるLabVIEWなどはコプロセッサがないと動きません．

計測環境を向上させる例として，3倍速OverDriveプロセッサの実装例について紹介します．

### ● 対象パソコン

筆者の手元には，CPUが486SX-25MHzのDOS/Vパソコン(PACKARD BELL社：PBL425S)があります．もちろんコプロセッサは付いていません．

パソコンの環境については以下のとおりです．

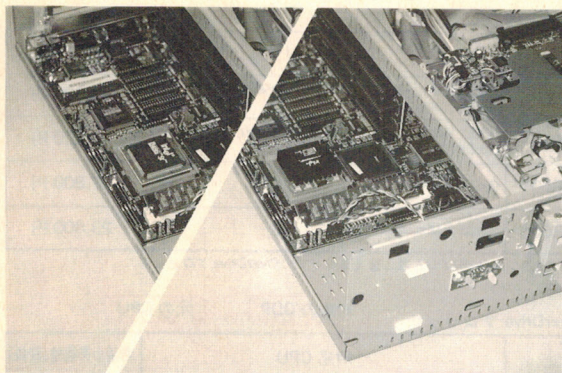
(1) RAM：8Mバイト
(2) ハードディスク：300Mバイト
(3) 解像度：1024×768 256色(1Mバイト)
(4) OS：Windows 3.1&OS/2
(5) モニタ：15インチ(飯山電機MF-8615)

特に，コプロセッサを必要としないアプリケーションに対してはこれでも支障はありません．しかし，アプリケーションによっては動作速度が遅くなることもありますが，これはいたしかたありません．

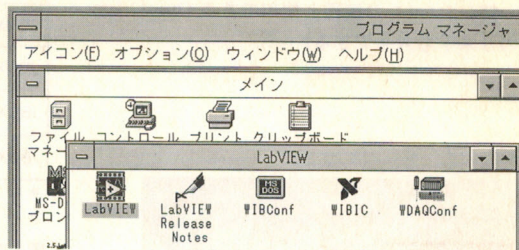
## 3倍速OverDriveプロセッサを搭載する

インテルOverDriveプロセッサ製品リスト(**表1**)から486SX 25MHz対応の3倍速OverDriveプロセッサ(以下，ODP)として，製品名JBOXDX4ODP75(**写真2**)を購入しました．

このODPはパソコン本体のCPUと差し替えて使う仕様になっています．元の486SX 25MHzと差し替えることによって，数値演算処理が3倍に高速化(25MHz×3＝75MHz)されます．

さらに，このODPにはコプロセッサが内蔵されていますので，浮動小数点演算はハードウェアで特に高

〈写真 3〉ODP の差し換え

(a) パソコン本体のカバーを開けたところ(写真左)
(486SX 25 MHz CPU が ZIF ソケットに搭載されている)
(b) ZIF ソケットに ODP(製品名 JBOXDX4ODP75)を
取り付けたところ(写真右)

〈図 2(a)〉計測制御ツール LabVIEW をインストールしたアイコン群

〈図 2(b)〉LabVIEW のフロントパネル(デフォルト)

速に処理されます.

また,この JBOXDX4ODP75 は元の CPU と単に差し替えるだけで,基本的にはパソコンのマザーボード上での設定変更などは必要ありません.ただし,念のためにパソコン購入店やメーカーに確認を取ることをおすすめします.

さっそく,CPU の差し替え作業をしましょう!

パソコン本体のカバーを開けます〔**写真 3**(a)〕.CPU(486SX 25 MHz)が搭載された ZIF ソケット(青色)を確認します.このソケットにはレバーが付いています.レバーをゆっくり引き上げながら CPU をはずします.慎重に作業してください.

こんどは ODP の差し込み作業です.ソケットの切り欠き部分と ODP の切り欠き部分を合わせるようにして,注意深く慎重に ODP を差し込み,レバーを元の位置までゆっくり押し下げます〔**写真 3**(b)〕.これで ODP の差し込みが完了しました.注意深く行えば作業そのものは難しいことはありません.

最後に,パソコン本体のカバーを取り付けます.

## LabVIEW をインストール

パソコンの電源を入れます.Windows の立ち上がりは ODP を搭載する以前に比べてかなり高速に感じます.必要であれば,再度,ビデオコントローラ画面で解像度などを確認します.

LabVIEW のプログラムディスク(Distribution Software 9 枚と Analysis Software 4 枚)を所定の方法でインストールします.なお,LabVIEW については以降の章で使用例などを解説していますので,興味ある方は参照してください.

無事,インストールが完了しました(コプロセッサがないとインストールの途中で“コプロセッサがないのでインストールができない”といった主旨のメッセージが出てくる).さっそく,LabVIEW を起動させてみましょう(**図 2**).LabVIEW での作業はなかなか快適です.

ODP を装着する前と装着後の動作速度について,比較データなどは特に取りませんでしたが,それほど“速い遅い”といったような感覚的な差異は感じられませんでした.また,486DX2(66 MHz)の DOS/V マシン(8 M バイトメモリ搭載)にも LabVIEW をインストールして比較しましたが,同じような使用感を得ることができました.

### ● おわりに

計測制御ツールにかぎらず CAE ツールといわれているものは,シミュレーションや数値計算がベースになります.都合により必要最少限度のパソコン環境で購入された方は,ODP を使ってパソコンのパワーアップを図ってみたらいかがでしょうか.もちろんメモリの増設なども並行して実施すれば,パソコン環境はより一層パワーアップすることが期待されます.

幸い,待望のペンティアム ODP(P24T)が 3 月に発売されました.P24T ソケット付きのパソコンを購入された方,この機会にぜひパワーアップを検討されてみてはいかがでしょうか.

なお,パソコンメーカーに事前確認することをお忘れなく!

# 第2章

データ収集カードをテスターの代わりに使うノウハウ

## データ収集用カードをパソコンで活用する方法

### ① PC/AT カードの活用

臼田昭司

ここでは，National Instruments 社[1]のデータ収集ボード（Multifunction I/O Board for the PC AT，型式：AT-MIO-16E-2，**写真1**）をテスター代わりに使った事例を説明します．このボードの本来の目的からすると，ちょっともったいない気もしますが，あまり手を加えずに“こんなこともできますよ！”といったたいへん簡単な計測例を紹介します．

## テスターとは

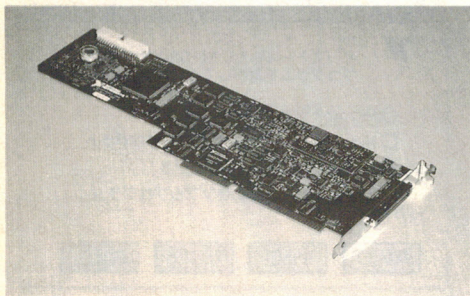
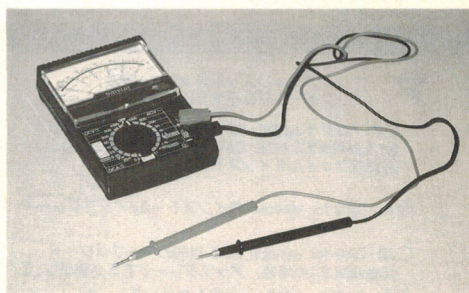
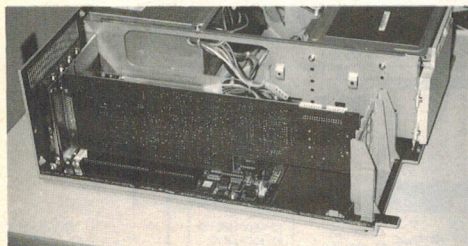
テスターとはどんなものか，ご存知でしょうか？ テスターと一口にいっても，いろいろな種類のものが市販されています（**写真2**）．

テスターとは簡単にいえば，2本のテストリード（先端はテストピン）を使って電圧を計ったり，電流を計ったり，そして抵抗値を計ったりするいわば“交直両用マルチメータ”です．指針表示（アナログ式）と数字表示（ディジタル式）があります．内蔵する電池も乾電池式と太陽電池式とがあります．許容誤差（測定精度を表す尺度）はだいたい±3％以内ですので高精度を要求しないのであれば，値をチェックしたり，確認したりするのには充分実用に足ります．

最近は，測定値を読みとるだけでなく，簡単な波形表示ができるタイプのテスターも発売されており，昨今，テスターのコンセプトもずいぶん変わってきました．

## データ収集ボードのインストールと環境設定

● **ハードウェアの設定**

最初にハードとソフトの準備です．データ収集ボード（A-D コンバータ）を DOS/V パソコン（AST 製

〈写真1〉 データ収集ボード（形式：AT-MIO-16E-2）

〈写真2〉 市販のテスター

〈写真3〉 データ収集ボードを DOS/V パソコンの拡張スロットに装着

〈写真4〉 68 ピンのシールドコネクタ・ブロック（形式：SCB-68）

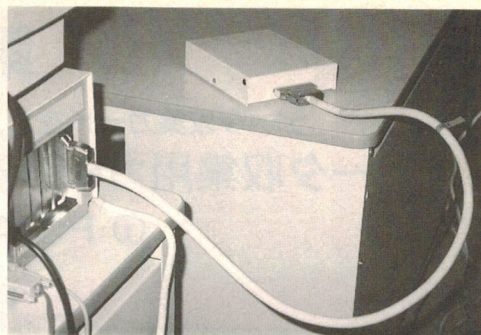
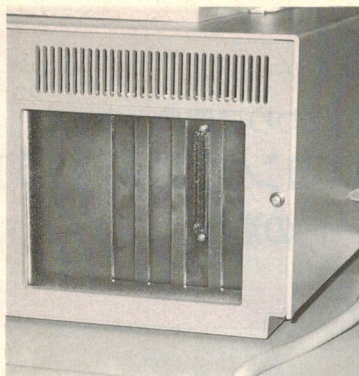
〈写真 5〉
データ収集ボードのI/Oコネクタ
(68 ピン)

BRAVO LC4/33)のカバーを開け，拡張スロットに装着します(**写真 3**)．ここではボード装着のみで，ボード上でのセッティングなどは一切ありません．

次に，テスターのテストピンに相当する 68 ピンのシールドコネクタ・ブロック(SCB-68，**写真 4**)と A-D コンバータの I/O コネクタ(68 ピン，**写真 5**)間を専用のシールドケーブル(SH6868，**写真 6**)で接続します．

これでハード面の準備は完了です．

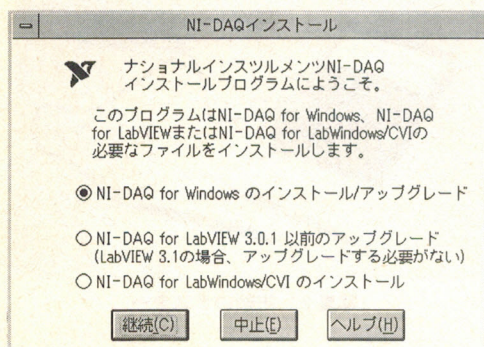
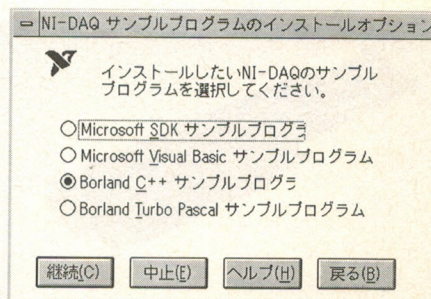
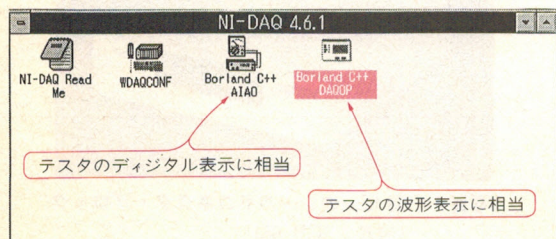
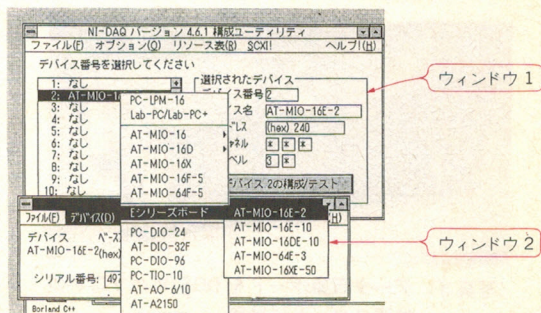
### ● ソフトウェアの設定

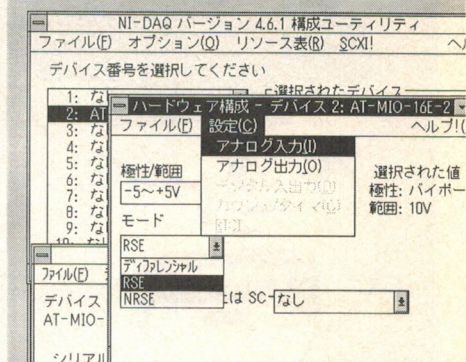
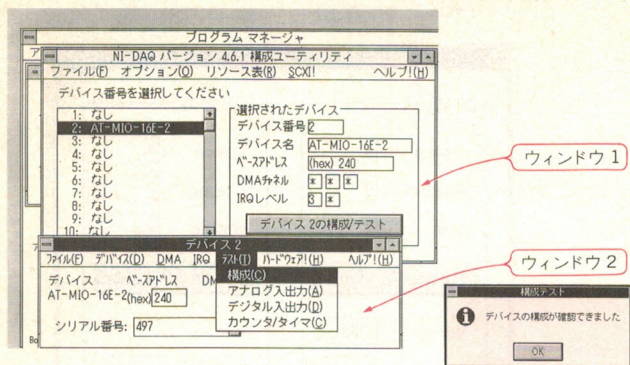
次は，ソフト面の準備です．付属のドライバソフト(NI-DAQ Ver.4.6.1)をインストールします．このドライバは DOS からでも Windows からでも，さらに

〈写真 6〉データ収集ボードのI/Oコネクタとシールドコネクタ・ブロックをシールドケーブルで接続したところ

LabVIEW[1]からでもインストールできます．ここでは Windows 上でインストールしました(**図 1**)．

インストールの途中で選択画面(**図 2**)がでてきますが，必要とするサンプルプログラムを選択してインストール完了です(**図 3**)．この図の Borland C++ AIAO と Borland C++ DAQOP のアイコン(サンプルプログラム)が，それぞれテスターの数字(ディジタル)表示と波形表示に相当します．アイコンの絵もそれらしい形をしています．

ここで，Borland C++のサンプルプログラムを選

〈図 1〉ドライバソフト(NI-DAQ)を Windows からインストールする

〈図 2〉サンプルプログラムの選択画面(インストールの途中)

〈図 3〉NI-DAQ の各アイコン(インストール完了)

〈図 4〉NI-DAQ 構成ユーティリティの各ウィンドウ

〈図5〉ウィンドウ2からテスト(T)の構成(C)を選択

〈図6〉デバイスの各入力条件を設定(標準設定)

択した理由は，たまたま筆者のパソコンには Borland C++ Ver.3.1 がインストールされていたためです．将来のアプリケーション拡張を考えてこれを選択しました．

● **環境設定**

次に，環境設定をします．**図3**のアイコン **WDAQ-CONF**(NI-DAQ 構成ユーティリティ)をダブルクリックすると**図4**のウィンドウ1が開きます．最初に，デバイス番号を「**2：なし**」をクリックします(筆者のパソコンでボードを差し込んだ拡張スロットのデバイス番号は2に設定されているため)．

〈写真7〉シールドコネクタ・ブロックの0チャネルにクリップ付きリード線を接続したところ

〈図7〉デバイスのアナログ入出力テスト結果

図中の「**デバイス2の構成/テスト**」ボタンをクリックします．ウィンドウ2が開きますから，**デバイス(D)**をクリックしてボードのタイプ一覧から拡張スロットに装着したボードのタイプ(AT-MIO-16E-2)選択します．その他の設定は自動で行います．

それでは，これらの設定が正しいかテストをしてみましょう．ウィンドウ2の**テスト(T)**をクリックして**構成(C)**を選びます(**図5**)．構成テスト結果はOKでした．

ここで，測定のための入力条件として，ウィンドウ2のハードウェア構成(極性/範囲，モード，アナログ入出力などの設定)を行います．標準的な設定をしました(**図6**)．なお，**図5**の**テスト(T)**の**アナログ入出力(A)**をクリックして，アナログ入力のアイコンを数回クリックしましたが，電圧出力はありませんでした(**Voltage＝0.000** を表示，**図7**)．これについては，68ピンシールドコネクタブロックにまだなにも入力を加えて

〈図8〉単三乾電池の電圧測定(ディジタル表示)

〈写真 8〉 発光ダイオードのアノードとカソード間の電圧測定

いませんので当然の結果です．

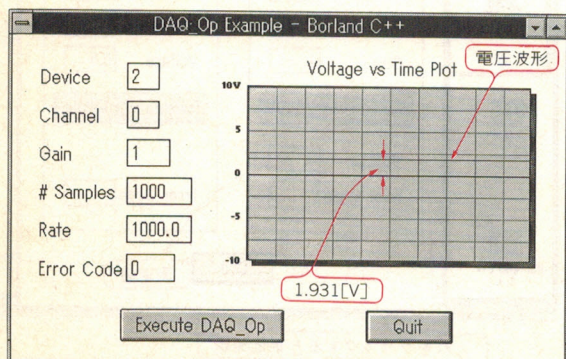
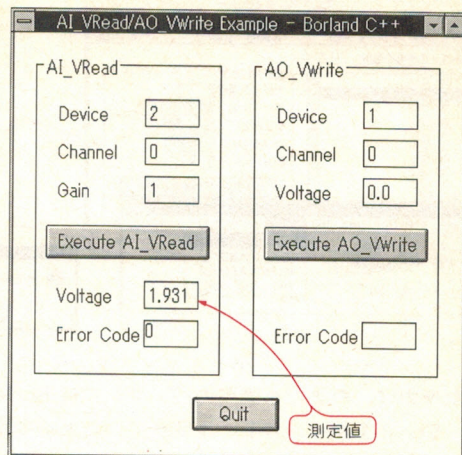
最後に，これらの設定をすべて保存して環境設定は終了です．

## データ収集ボードで直流・交流を計る

テスターと比較しながら，簡単な直流と交流測定について行ってみます．

**(1) 直流電圧を計る**

手元に，単三乾電池が 2 個直列に電池ホルダーに入っています．これの電圧を**写真 2** のテスターで計ります．おおよそ 3.2[V] です．こんどはデータ収集ボードです．68 ピンのシールドコネクタ・ブロックの 0 チャネル(68 ピンと 24 ピンを使用，**写真 7**)に適当なクリップ付きリード線を接続して電池ホルダーの＋/－側をクリップします．**図 3** のアイコン **Borland C++ AIAO** をダブルクリックして測定開始です．**図 8** の **Execute AI_VRead** ボタンをクリックすると，**Voltage** の表示ボックスに 3.179[V]とディジタル表示されました．

〈図10〉 発光ダイオードのアノード-カソード間電圧の測定（波形表示）

〈図 9〉 発光ダイオードのアノード-カソード間電圧の測定（ディジタル表示）

もうひとつ計りましょう．発光ダイオード(LED)のアノード-カソード間電圧です(**写真 8**)．上の電池ホルダーの＋/－端子に抵抗(330 Ω)を介して LED のアノードとカソードを接続します．テスターでは 1.93[V] ぐらいでしょうか．それではアノード，カソードをクリップして **Execute AI_VRead** ボタンをクリックします．**Voltage** 表示は 1.931[V]となりました(**図 9**)．

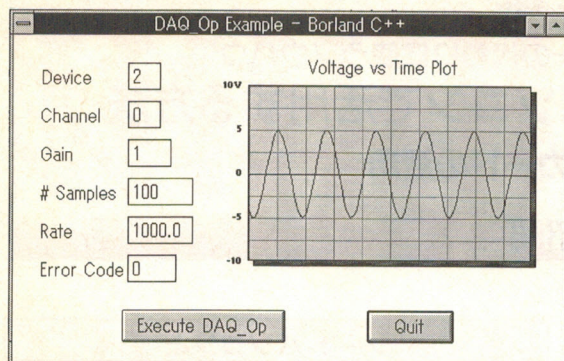
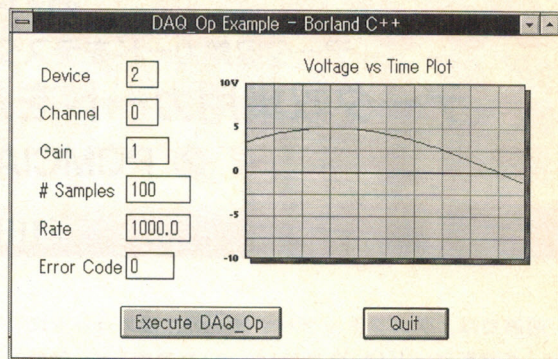
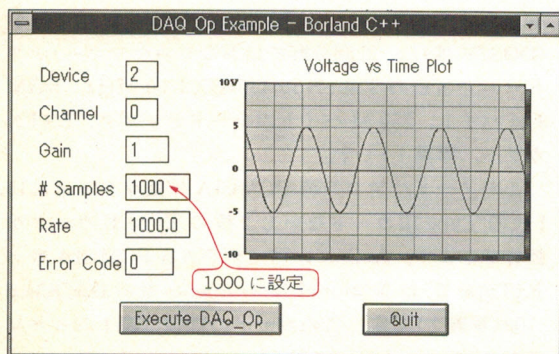
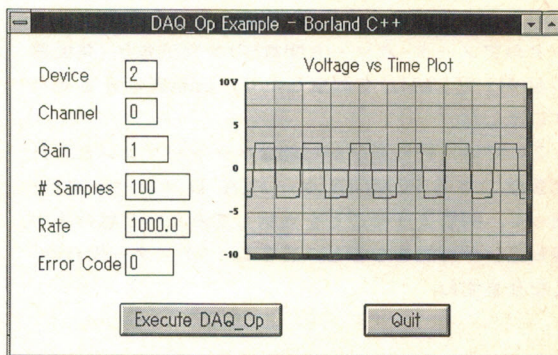
テスター代わりとしてこんな具合に使えます．

それでは最近の波形表示付きのテスターのつもりで，**図 3** の **Borland C++ DAQOP** をダブルクリックします．LED のアノード-カソード間電圧をそのまま計ります．**Execute DAQ_Op** ボタンをクリックすると，**Voltage vs Time Plot** という波形表示画面に 1 本の電圧線が表示されました(**図10**)．直流電圧ですからこのようになるのは当然です．

**(2) 交流電圧を計る**

ここに発振器がありましたので，適当な交流電圧を出してこれを計測してみます(**写真 9**)．発振器の周波数を 60[Hz] に設定します．発振器の出力電圧につい

〈写真 9〉 発振器を使って交流電圧を測定

〈図11〉 交流電圧(最大値5〔V〕，周波数60〔Hz〕)の波形表示

〈図12〉 高周波電圧(周波数1〔kHz〕)の波形表示 (#Samples：100の場合)

〈図13〉 高周波電圧(周波数1〔kHz〕)の波形表示 (#Samples：1000の場合)

〈図14〉 矩形波(周波数60〔Hz〕)の波形表示

ては，テスターの電圧表示がだいたい3.54[V]（実効値)になるように調整します．それでは波形表示画面をみてみましょう(**図11**)．サイン波形です．ピーク電圧(最大値)は5[V] です．

実効値×$\sqrt{2}$(1.414)＝最大値

の関係が成り立ちます．

もう一つ測定をします．高周波電圧(1[kHz]の場合)を見てみましょう．**図12**のようになりました．**#Samples**の**100**を**1000**にして波形を見やすくします．**図13**は1[KHz] の高周波電圧です．サイン波形です．

こんどは波形形状を変えてみます．矩形波です．**図14**は60[Hz] の場合です．

● **おわりに**

データ収集ボードを用いて，テスターのような感覚で，直流と交流について簡単な電圧測定と波形表示についてご紹介しました．この計測ボードは，アナログ，ディジタル，タイミングI/O機能をもった“All-in-one”のボードです．また，ドライバとしてのソフトについてもLabVIEWなどと組み合わせることによりこのボードの優れた特徴が最大限に引き出せます．

機会がありましたなら一度お試しください．

**国内リリース元**

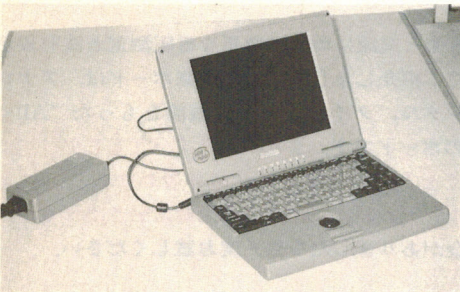
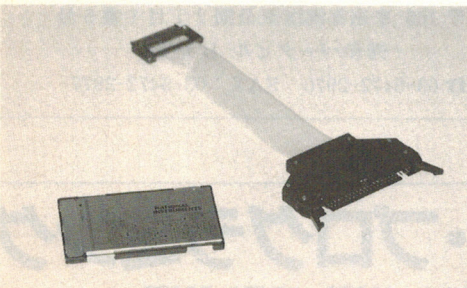
(1) 日本ナショナルインスツルメンツ(株)
〒105 東京都港区芝公園2丁目1番5号
秀和パークビル B館5F
☎ 03-5472-2970　FAX：03-5472-2977

# 第2章

## データ収集カードをテスターの代わりに使うノウハウ
# データ収集用カードをパソコンで活用する方法
## ② PCMCIAカードの活用

臼田昭司

前節では，デスクトップパソコンの拡張スロットにデータ収集ボード(A-Dコンバータ)を装着して，付属のドライバソフト(NI-DAQ Ver.4.6.1)をインストールした環境を用意しました．そして，このソフトに内蔵されていたサンプルプログラムを使って，テスターのような感覚で直流電圧(電池の電圧や発光ダイオードのアノード-カソード間電圧)や交流電圧(発振器の60 Hzや1 kHz)を測定したり，波形表示をさせたりしました．

ここでは，持ち運びが可能なノートパソコンとこれに装着できるPCMCIAカード(A-Dコンバータ)を使って，同じようなことを実験してみたいと思います．持ち運びが可能となると，まさしく“テスター代わり”となりますね．

〈写真1〉ソニー・テクトロニクス社のTake Note 486

〈写真2〉National Instruments社のDAQCard-700と付属のI/Oケーブル

### ノートパソコンとPCMCIAカードの紹介

さて，最近のノートパソコンもひところと違い，いろいろな機能が搭載されています．PCMCIA拡張ソケットもその一つです．今回使用したノートパソコン(DOS/V機)は，計測分野ではおなじみのソニー・テクトロニクス社のTakeNote486(DX2-50 MHz，RAM 8 Mバイト，250 Mバイトのハードディスク，DSTNカラー，**写真1**)です．

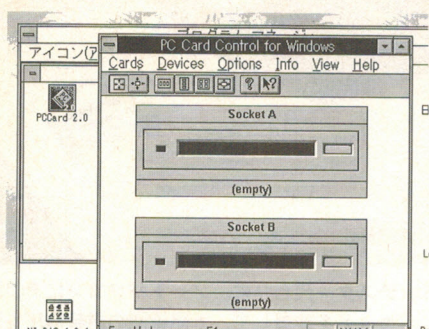
このノートパソコンのPCMCIA拡張ソケットには，1枚のType IIIカードないし2枚のType IIカードが装着できます．A-DコンバータであるPCMCIAカード(Type II)はNational Instruments社のDAQCard-700(**写真2**)です．このカードの仕様は**表1**のとおりです．

このカードには，カードの使用状態を表示するユーティリティプログラム(一般にはカードサービスという)CardWare 2.0が標準で付いています．また，ドライバソフトNI-DAQ Ver.4.6.1も付属しています．

前の節で紹介したハード/ソフトと異なるところは，デスクトップパソコンがノートパソコンに，A-Dコンバータがボードからカードに，そしてカードには新た

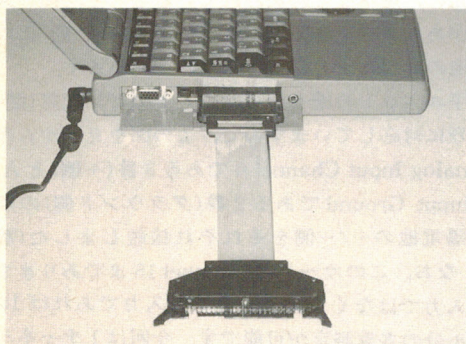
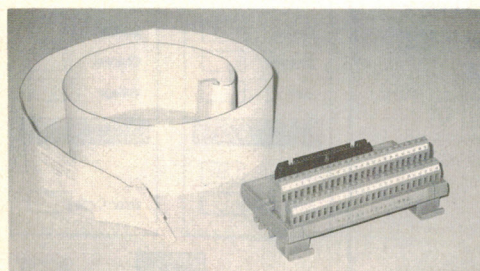
〈表1〉DAQCard-700の概略仕様

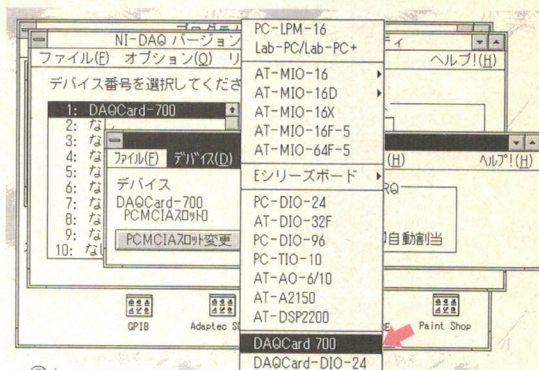
| | |
|---|---|
| 12ビットA-Dコンバータ | ▶16の多重化されたシングルエンド入力<br>▶100,000サンプル/秒のサンプリングレート<br>▶多チャネルスキャン<br>▶入力範囲：±10 V，±5 V，±2.5 V<br>▶A-D変換タイミング用タイマ<br>▶16サンプルFIFO A-D変換バッファ<br>▶割り込み発生<br>▶内部/外部A-D変換タイミング |
| 省電力 | +5 $V_{DC}$で120 mA |
| ディジタル入出力 | 8ビット入力，8ビット出力ポートとして構成される16ディジタルI/Oライン |
| カウンタ/タイマ | 周波数計測，イベントカウンティング，パルス出力アプリケーションのための2基の16ビットカウンタ/タイマ |
| 割り込み | 一つのカウンタ出力への割り込み発生 |
| 対応ソフト | LabVIEW，LabWindows/CVI，LabWindowsにてプログラム可能 |

〈図1〉カードサービス(Card Ware 2.0)の起動画面．empty(PCMCIAカードがまだ装置されていない)状態を示す

にカードサービスが付いたということでしょうか．ドライバソフトは同じものです．

## カードサービスとドライバソフトのインストール

### ● カードサービス

マニュアルに従ってカードサービスをインストールします．特に難しいところはありません．インストールの途中でいくつかの選択項目がありますが，デフォルトをそのまま選択します．インストールがうまくいけば，Windowsを再起動すると**図1**のようなカードの使用状態を示す表示が現れます．まるでPCMCIA

〈写真3〉DAQCard-700をTake Note 486の拡張ソケットに差し込む

〈写真4〉延長I/Oケーブルと端子ユニット

〈図2〉ドライバソフトNI-DAQ Ver.4.6.1でデバイス(D)のタイプ(DAQCard-700)を選択

拡張ソケットを絵にしたようですね．

この図を見れば，Socket　AもSocket　Bも空(empty)になっています．要するに，カードがまだ差し込まれていないということを示しています(実際にまだカードを装着していない)．

なお，通常はPCMCIA拡張ソケット付きのノートパソコンにはカードサービスがプリインストールされています．TakeNote486にはPCMCIAユーティリティ・フロッピディスクが付属しています．

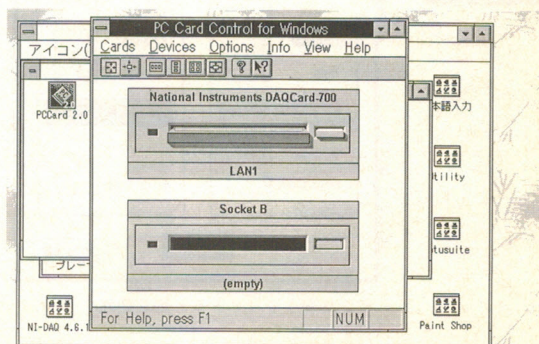
**図1**のコントロールメニュー・ボックスから「閉じる(ト)」をクリックしてカードサービスを終了させます．

### ● ドライバソフト

ドライバソフトNI-DAQ Ver.4.6.1のインストールについては，前の節で説明した方法と同じです．ただし，デバイス(D)のタイプを選択する画面では，DAQCard-700を選択します(**図2**)．

## PCMCIAカードのセットアップ

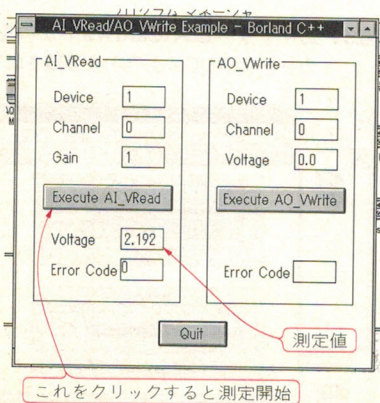
最初に，PCMCIAカードと付属のI/Oケーブルを

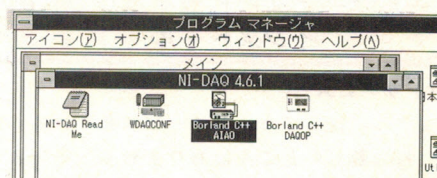
〈図3〉カードサービス(CardWare 2.0)を起動．PCMCIAカード(DAQCard-700)が装着されている状態を示す

〈図 4〉 DAQCard-700 の I/O コネクタのピン配置

接続します．次に，カードをノートパソコンの拡張ソケットの上段(Socket A に対応)に注意深く差し込みます(**写真 3**)．ケーブルの接続方向やカードの差し込み方向など指定がありますので注意が必要です．

なお，付属の I/O ケーブルは長さが短く，ケーブル先端もこのままでは測定端として不便なので，あらたに延長ケーブルと端子ユニットを準備しました(**写真 4**)．これらはオプションとして用意されています．

PCMCIA カードを装着したところで，CardWare 2.0 を起動させてみましょう(**図 3**)．確かに，Socket A のほうにカード(National Instruments DAQCard_700)が装着されています．表示も empty から LAN 1 に変わっています．これで確認できました．

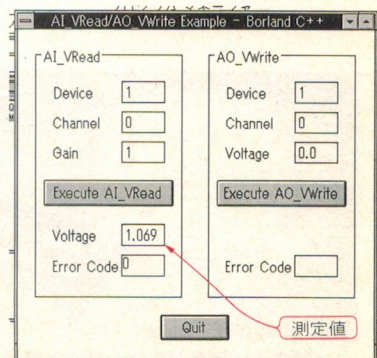
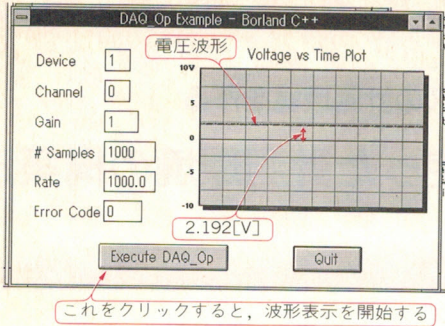
**〈図 6〉 太陽電池の誘起電圧を測定**

**〈写真 5〉 テスター代わりに，ノートパソコンに PCMCIA カードを装着して，太陽電池の誘起電圧を測定**

**〈図 5〉 ドライバソフト NI-DAQ Ver.4.6.1 を起動しディジタル表示(Borland C++ AIAO)アイコンを選択**

## 太陽電池の電圧を計る

手元に，電卓などに使われているものと同種の太陽電池があります．大きさは約 35×15 mm です．この太陽電池の誘起電圧を計ってみたいと思います．

端子ユニットの端子番号はカードのピン配置(**図 4**)の番号に対応していますので，この図を見ながら今回は Analog Input Channel 0 である 3 番(＋側)と Analog Input Ground である 2 番(グラウンド側)の端子に太陽電池の＋/－側をそれぞれ接続しました(**写真 5**)．なお，このカードは Channel 15 まであります．差動入力ではなくシングルエンド入力であれば 16 チャネル分の多重測定が可能です．今回は 1 チャネル分

**〈図 7〉 太陽電池を手でふさいだときの誘起電圧を測定**

〈図8〉太陽電池の誘起電圧の電圧波形

〈図9〉小型電源トランスの二次側(交流電圧9〔V〕)の電圧波形

のみの測定です．

さっそく測定してみましょう．NI-DAQ Ver.4.6.1を起動させます．ディジタル表示(Borland C++ AIAO)と波形表示(Borland C++ DAQOP)アイコンからディジタル表示を選びます(**図5**)．

Execute AI_Vreadボタンをクリックすると，太陽電池の電圧は2.192[V]と出ました(**図6**)．太陽電池を手でちょっとふさいでみます．電圧はいくらになるかな！ 1.069[V]となりました(**図7**)．太陽電池への入射光をさえぎりましたので当然の結果です．手持ちのテスター(アナログ表示，前の節の**写真2**)ではどうでしょう．えーっと…約1.0[V]です．ディジタル表示のほうがアナログ表示より便利ですね！

次に，波形表示も見てみましょう．アイコンBorland C++ DAQOPをダブルクリックします．表示結果は**図8**です．太陽電池の電圧は直流ですから一本の線として表示されました．

こんな具合です．なお，この辺りのNI-DAQの使い方は前の節の場合と同じです．

## 交流電圧波形を見る

一般家庭のコンセントなどにきている電圧は50 Hzないし60 Hzの100 Vの電圧(実効値)です．波形はサイン波形(正弦波形ともいう)です．これを見てみましょう．

といっても，100 Vの電圧を直接計るのは危険ですから小型電源トランス(一次100[V]：二次9[V])を使って低電圧(9 V)に落とします(**写真6**)．アイコンBorland C++ DAQOPをクリックして起動させます．波形表示は**図9**のようになりました．#Samplesはデフォルトは1000ですが，波形を見やすく100に設定しました．確かにサイン波形です！

### ● おわりに

PCMCIAカード(A-Dコンバータ)とノートパソコンを使った計測体験は以上のとおりです．前の節で紹

〈写真6〉テスター代わりに，PCMCIAカードをノートパソコンに装着して，小型電源トランスの二次側電圧波形を見る

介したデータ収集ボードの場合と基本的には同じです．ノートパソコンであれば持ち運びに便利で，ボードの場合とは違った応用例が期待できそうです．

最後に，PCMCIAカードの導入に際しては，PCMCIAのメーカーないしはノートパソコンの機種を確認する必要があります．PCMCIA拡張ソケットが付いているからどのPCMCIAカードでも使えるというわけにはいかないようです．ノートパソコンの機種によってはPCMCIAカードがうまく動作しないこともあるようです．

**問い合わせ先**

(1) ソニー・テクトロニクス㈱
　ソリューションシステム部
　〒141 東京都品川区北品川5-9-31
　☎03-3448-4851

**◆ 参考文献 ◆**

(1) DAQCard-700 Users Manual, National Instruments Corp..

# 第3章

## ノートパソコンによる省スペースの計測環境を作る
# GPIB カード(PCMCIA)を使った計測環境の構築法

臼田昭司

## GPIB カードと周辺機器

ここで用いる GPIB カードは，JEIDA 4.2/PCMCIA 2.1 に準拠した計測・制御用の PC カードです．この種のカードタイプはまだそうたくさんは市販されていませんが，最近になって少しずつ関連のメーカーから市販されるようになってきました．DOS/V 機用もいくつか製品化されてきました．

ここでは，アドテックシステムサイエンス社[1]製の DOS/V 機用 GPIB カード(型式：AXP-GP01)を使った簡単な計測について紹介します．このカードは，TYPE II(厚さ：5.0 mm)の規格に準拠しています(**写真 1**)．

使用した DOS/V 機は，IBM 製のノートパソコン ThinkPad230Cs です．この DOS/V 機には TYPE II のスロットが 2 個備えられています．もちろん，PC カードの機能を制御するカードサービス(ソフト名：EasyGoing)がプリインストールされています(**図 1**)．

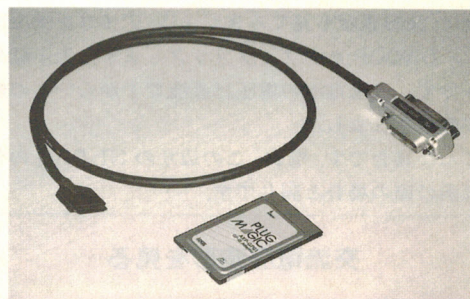
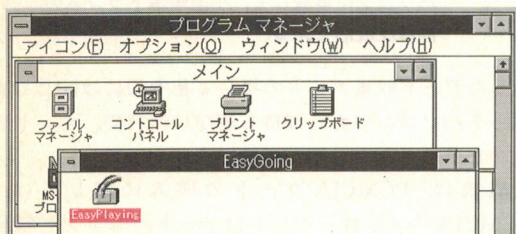
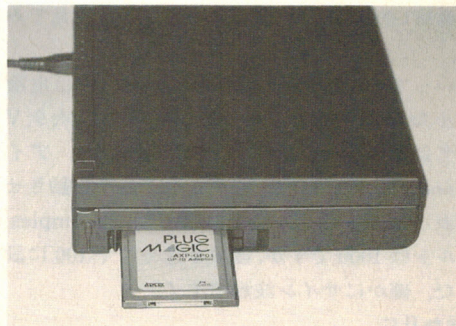
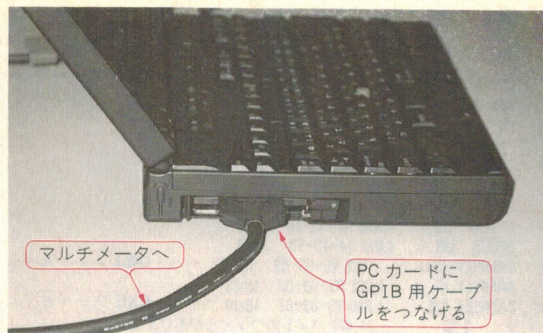
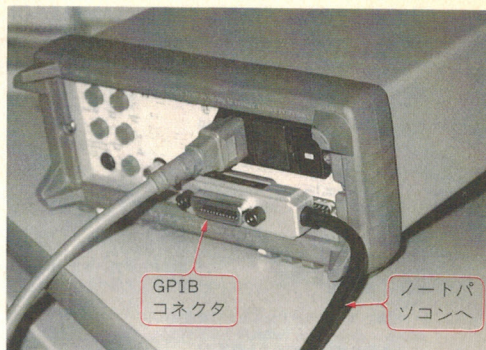
次に，計測を行うためには測定器が必要ですが，HewlettPackard(HP)社のマルチメータ HP34401A(**写真 2**)を使って簡単な電圧測定をしてみます．このマルチメータは，この世界でいわば標準的機器になっている SCPI, HP3478A, Fluke8840A のプログラミング言語が設定できるようになっています．

## ハードとソフトのインストール

最初に，ハードウェアのインストールをします．

GPIB カードを ThinkPad230Cs の 2 個あるスロッ

〈図 1〉 ThinkPad230Cs のカードサービス EasyGoing

〈写真 2〉 測定に用いたマルチメータ(HP 社 HP34401A)

〈写真 1〉 GPIB カード(AXP-GP01)と付属接続ケーブル

〈写真 3〉 GPIB カードをノートパソコン(IBMThinkPad 230Cs)のカードスロットに装着しているところ

〈写真 4(a)〉付属ケーブルをノートパソコンに装着したGPIB カードに接続

〈写真 4(b)〉付属ケーブルをマルチメータ背面のGPIB コネクタに接続

トの内，上のスロットに装着します(**写真 3**)．次に，GPIB カード付属の接続ケーブルで，指定された方向にGPIB カードとマルチメータ(背面)を注意深く接続します(**写真 4**)．

次に，ソフトウェアのインストールをします．

GPIB デバイスドライバ(**AXPGPIB**)をパソコン側に組み込みます．付属のフロッピをAドライブに差し込んで，

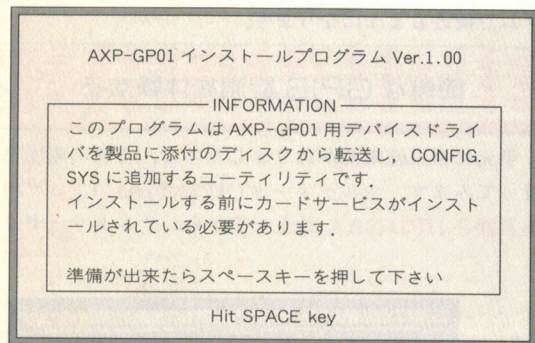
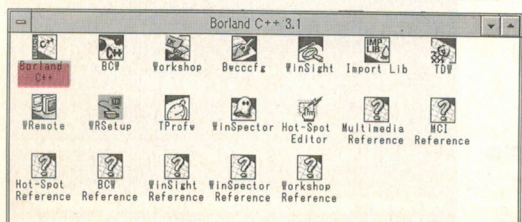
**A:¥>inst**(リターン)

とします(**図 2**)．あとは画面の指示どおり行えばインストール完了です．完了後は**CONFIG.SYS** ファイルに，

**DEVICE＝C:¥AXPGPIB¥AXPGPIB.SYS**

を追加します(**図 3**)．

〈図 2〉GPIB デバイスドライバ(AXPGPIB)のインストール画面

```
AXP-GP01 インストールプログラム Ver.1.00

INFORMATION
このプログラムは AXP-GP01 用デバイスドライ
バを製品に添付のディスクから転送し，CONFIG.
SYS に追加するユーティリティです．
インストールする前にカードサービスがインスト
ールされている必要があります．

準備が出来たらスペースキーを押して下さい

Hit SPACE key
```

〈図 4〉Borland C++ Ver.3.1 の Windows 画面

なお，今回計測で使うプログラムのコンパイラとして，Borland 社の C＋＋ Ver 3.1 を使いました．これもあらかじめインストールしておきます．C＋＋は，DOS からでも Windows(**図 4**)からでも立ち上げることができます．

## サンプルプログラムとコンパイル

付属のフロッピディスクにはデバイスドライバのほかに，二つのサンプルプログラムとこれをコンパイルするためのライブラリ(スモールモデル **GPSLIB.LIB** とラージモデル **GPLLIB.LIB**)，さらに GPIB ライブラリ用インクルードファイル(**AXPGP.H**)などが入っています．

二つのサンプルプログラムは簡単な実験であればすぐにでも使える実用的なものです．ここでは，『マルチ

〈図 3〉CONFIG.SYS ファイルに GPIB カードドライバ AXPGPIB.SYS を追加

```
BUFFERS=20
FILES=30
DOS=HIGH,UMB
COUNTRY=081,932,C:¥DOS¥COUNTRY.SYS
SHELL=C:¥DOS¥COMMAND.COM /P /E:512
DEVICE=C:¥DOS¥$FONT.SYS
DEVICE=C:¥DOS¥HIMEM.SYS
DEVICE=C:¥WINDOWS¥EMM386.EXE RAM X=C800-CFFF
DEVICEHIGH=C:¥DOS¥SETVER.EXE
DEVICEHIGH=C:¥DOS¥$DISP.SYS
DEVICEHIGH=C:¥DOS¥$IAS.SYS
DEVICE=C:¥EZGOING¥IBMDSS01.SYS /M
DEVICE=C:¥EZGOING¥IBMDOSCS.SYS
DEVICE=C:¥EZGOING¥DICRMU01.SYS /MA=C800-CFFF
DEVICE=C:¥EZGOING¥$ICPMDOS.SYS
DEVICEHIGH=C:¥DOS¥POWER.EXE
DEVICE=C:¥EZGOING¥AUTODRV.SYS C:¥EZGOING¥AUTODRV
.INI
DEVICE=C:¥EZGOING¥DISKDRV.SYS
DEVICEHIGH=C:¥DOS¥$PRN.SYS
DEVICEHIGH=C:¥DOS¥ANSI.SYS /X
DEVICEHIGH=C:¥DOS¥$FDD12.SYS
rem DEVICEHIGH=C:¥DOS¥RAMDRIVE.SYS
INSTALL=C:¥DOS¥IBMMKKV.EXE /M=S /Z=4 /C /L /J=90
 /S=C:¥DOS¥MULTDICT.PRO /U=C:¥$USRDICT.DCT
STACKS=9,256


DEVICE=C:¥AXPGPIB¥AXPGPIB.SYS  ← 追加されたカードドライバ
```

〈リスト 1〉 サンプルプログラム(SAMPLE1.C)

```
/*
        AXP-GP01 C言語ライブラリ サンプルプログラム－1
          データの送受信
        1994  ADTEK SYSTEM SCIENCE Co.,Ltd.
*/

#include        <stdio.h>
#include        <string.h>                                  (1)
#include        <AXPGP.h>  ←              /* AXPGP01ライブラリ用ヘッダ      */

void    main( void ){                                       (2)
        int     len, cnt, status ;
        char    send_buf[10] ; ←          /* 送信データ用バッファ(ｺﾏﾝﾄﾞ)  */
        char    buf[20] ;                 /* 受信データ用バッファ          */

        /* 初期化 カードのアドレスを0に設定 */
        status = gpib_init( (uchar)0 ) ; ←                  (3)
        /* 返り値が NORMAL 以外なら終了 */
        if( status != NORMAL )  goto    _EXT ;

        /* IFC 送信 */
        status = gpib_ifc() ; ←                             (4)

        /* REN イネーブル */
        gpib_set_ren( SET_TRUE ) ; ←                        (5)

        /*** HP3478A にコマンドを送信する ***/
        strcpy( send_buf, "F1R0N4" ) ;

        /* コマンド文字列長を求める */
        len = strlen( send_buf ) ;

        /* 送信           */
        gpib_send( (uchar)3, len, send_buf, DELIM_CRLF ) ; ←   (6)

        /* 100回読み込む */
        for( cnt = 0 ; cnt < 100 ; cnt++){

                /* 読み込むﾃﾞｰﾀﾌｫｰﾏｯﾄ : +9.99999E+9CRLF (13バイト)    */
                gpib_recv( (uchar)3, 14, buf, DELIM_CRLF ) ; ←  (7)
                buf[11] = '¥0' ;

                /* 読み込んだデータを画面表示 */
                printf( "[%03d] : [%s]¥n", cnt, buf ) ;
        }

        /* REN ディスエーブル */
        status = gpib_set_ren( SET_FALSE ) ; ←              (8)

_EXT:
        /* GPIB の終了 */
        gpib_close() ; ←                                    (9)

}
```

メータからのデータを読み込んで，マルチメータのディスプレイ表示と同じ値をパソコンのディスプレイに表示する(以下，GPIB 計測とする)』プログラム **SAMPLE1.C** を使うことにしました．

**リスト 1** は **SAMPLE1.C** の中身です．このプログラムの詳細は省略しますが，プログラム中の(1)，(2)，(3)，(4)，(5)，(6)，(8)，(9)のマーキングは，すべて GPIB カード AXP-GP01 専用のライブラリ関数です

〈図 5〉 サンプルプログラム(SAMPLE1.C)をコンパイルして OBJ ファイルと EXE ファイルを作成

```
A:¥>dir

 ドライブ A にはボリューム・ラベルがありません
 ボリューム・シリアル番号は 15F0-040F です
 ディレクトリーは  A:¥

GPSLIB   LIB       4,096 94-10-13    1:00
GPLLIB   LIB       4,096 94-10-13    1:00
SAMPLE1  OBJ         825 95-03-02   18:19   ← OBJ ファイル
SAMPLE1  C         1,222 95-03-02   13:22
SAMPLE1  EXE      11,240 95-03-02   18:19   ← EXE ファイル
        5 個          21,479 バイトのファイルがあります
                   1,435,648 バイトが使用可能です
```

これらの関数についてはマニュアルに詳しく説明されています．

● **コンパイル**

最初に，新しいフロッピディスクを1枚用意します．このフロッピに付属ディスクからサンプルプログラム(**SAMPLE1.C**)とライブラリ(**GPSLIB.LIB** と **GPLLIB.LIB**)をコピーします．

ノートパソコン(ThinkPad230Cs)を起動し，DOS 画面にします．ここで，ドライブを替えます．

**C:¥>A:**(リターン)

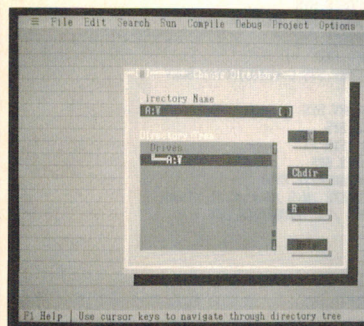
**A:¥>**

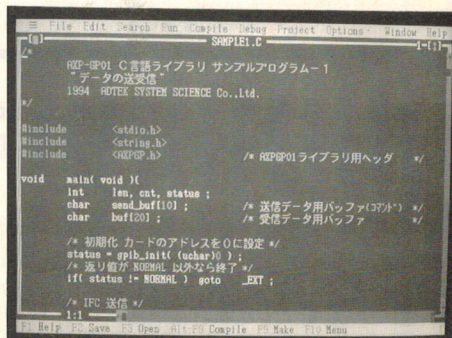
新しいフロッピを A:ドライブに入れて，次のようにして **SAMPLE1.C** をコンパイルします．

**A:¥>BCC -ml SAMPLE1.C GPLLIB.LIB**(リターン)

ラージモデルでコンパイルしました(スモールモデルでもかまわない)．コンパイルの結果，フロッピには **EXE** ファイルが作られました(**図 5**)．このコンパイルとリンクによって，先ほどのライブラリ関数がプログラムで使えるようになります．

## 簡単な GPIB 計測を体験する

手元に単三乾電池がありましたので，これの電圧を計ってみます．マルチメータ(HP34401A)のプログラム言語を HP3478A に設定してから，テストリードを

◀〈写真 5〉 立ち上がった Borland C++のディレクトリをC:¥からA:¥に設定(LCD の画面のコンストラストが低かったので，CRT 画面を写真にとっている)

▶〈写真 6〉 サンプルプログラム(SAMPLE1.C)をオープンしたところ

〈図6〉単三乾電池の電圧測定の場合のノートパソコンが読み込み表示したデータ例

```
A:¥>sample1
[000] : [+1.59330E+0]
[001] : [+1.59330E+0]
[002] : [+1.59330E+0]
[003] : [+1.59330E+0]
[004] : [+1.59330E+0]
[005] : [+1.59330E+0]
[006] : [+1.59330E+0]
[007] : [+1.59330E+0]
[008] : [+1.59330E+0]
[009] : [+1.59330E+0]
[010] : [+1.59330E+0]
[011] : [+1.59320E+0]
[012] : [+1.59330E+0]
[013] : [+1.59330E+0]
[014] : [+1.59330E+0]
```

(a)

```
[078] : [+1.59330E+0]
[079] : [+1.59330E+0]
[080] : [+1.59330E+0]
[081] : [+1.59330E+0]
[082] : [+1.59330E+0]
[083] : [+1.59320E+0]
[084] : [+1.59330E+0]
[085] : [+1.59330E+0]
[086] : [+1.59330E+0]
[087] : [+1.59330E+0]
[088] : [+1.59330E+0]
[089] : [+1.59330E+0]
[090] : [+1.59330E+0]
[091] : [+1.59330E+0]
[092] : [+1.59330E+0]
[093] : [+1.59330E+0]
[094] : [+1.59330E+0]
```

(b)

電池の＋/－側にクリップします．マルチメータの電圧表示は1.593Vです．はたしてパソコン画面にも同じ値が表示されるでしょうか．

DOSからでもWindowsからでもかまいませんが，ここではWindowsからC++を立ち上げることにします．**図4**の左上のBorland C++のアイコンをダブルクリックします．

メニューの**File**の中の**Change dir...**でディレクトリを**A:¥**に設定します(**写真5**)．次に，**File**の**Open..**からプログラム**SAMPLE1.C**をオープンします(**写真6**)．コンパイルはすでに前の項で済んでいますので，ここではメニューの**Run**のみを実行させます．

実は，**リスト1**のプログラムは100回データを読み込んで，それをそのまま表示する**for**文を使っていますので，実行結果としては100個のデータ列が得られました．**図6**はその一部です．データ番号[**000**]から[**099**]まで100個データ表示されます．これらはマルチメータの表示と一致しました．

次に，単三乾電池を2個直列にして，330Ωの抵抗$R$を介して発光ダイオード(LED)を点灯させたときの電流$I$を計ってみます．電流は抵抗(330Ω)の両端電圧$V$から，$I=V/R$として求めます(マルチメータは電流を直接計ることができますが，ここではあえてこのようにしました)．電池ホルダー，LED，抵抗については，基板上に配置して接続しました．

準備完了です(**写真7**)．マルチメータの表示は1.225Vです．電流を計算すると，$I=1.225/330=0.0037$ Aとなります．さて，パソコン側ではどうでしょう．ファイルメニューから**Run**をクリックして実行です．結果は，**図7**です．ほとんど一致した結果が得られました．

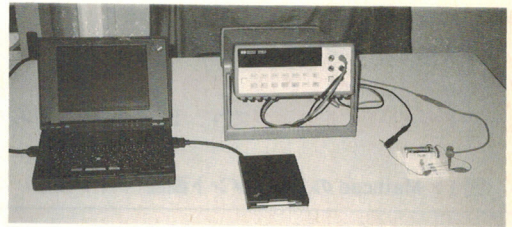
こんな具合です．実際やってみるとなかなかおもしろい体験です．GPIB計測の簡単な実験です．

〈図7〉発光ダイオードに流れる電流(抵抗の両端電圧)測定の場合のノートパソコンが読み込み表示したデータ例

```
A:¥>sample1
[000] : [+1.22190E+0]
[001] : [+1.22190E+0]
[002] : [+1.22190E+0]
[003] : [+1.22190E+0]
[004] : [+1.22190E+0]
[005] : [+1.22200E+0]
[006] : [+1.22190E+0]
[007] : [+1.22200E+0]
[008] : [+1.22190E+0]
[009] : [+1.22190E+0]
[010] : [+1.22190E+0]
[011] : [+1.22190E+0]
[012] : [+1.22200E+0]
[013] : [+1.22200E+0]
[014] : [+1.22190E+0]
```

(a)

```
[078] : [+1.22420E+0]
[079] : [+1.22420E+0]
[080] : [+1.22420E+0]
[081] : [+1.22420E+0]
[082] : [+1.22420E+0]
[083] : [+1.22420E+0]
[084] : [+1.22430E+0]
[085] : [+1.22420E+0]
[086] : [+1.22420E+0]
[087] : [+1.22430E+0]
[088] : [+1.22420E+0]
[089] : [+1.22420E+0]
[090] : [+1.22430E+0]
[091] : [+1.22430E+0]
[092] : [+1.22420E+0]
[093] : [+1.22420E+0]
[094] : [+1.22430E+0]
```

(b)

〈写真7〉発光ダイオードに流れる電流(抵抗の両端電圧)をGPIBシステムで測定している全景

## 計測制御用カードとノートパソコン

GPIBやA-Dコンバータなど計測制御用のカードは使うノートパソコン(特に，DOS/V機)の機種などによって，カードに付属しているドライバがうまく起動できなかったり，カードサービスも専用のものが付いてきたりして，どれを優先して使っていいものか迷うこともあります．使うノートパソコンによっては動かないものもあると聞いています．購入する場合には，このあたりをよく確認する必要があります．

この種のカードは商品化されたばかりで，サポートするノートパソコンのバリエーションもまだまだ少ない状況です．今後はより多くのノートパソコンに対応したカードが増えてくるものと思います．

**本稿で使用したGPIBカードの問い合わせ先：**

(1) (株)アドテック システム サイエンス
〒240 横浜市保土ヶ谷区天王寺町1-16-6
☎045-331-7575　(FAX)045-331-7770

# 第4章

## LaBDAQによるDOS環境の計測システム

# PC98用計測・解析ソフトを使ったFFT解析の方法

臼田昭司

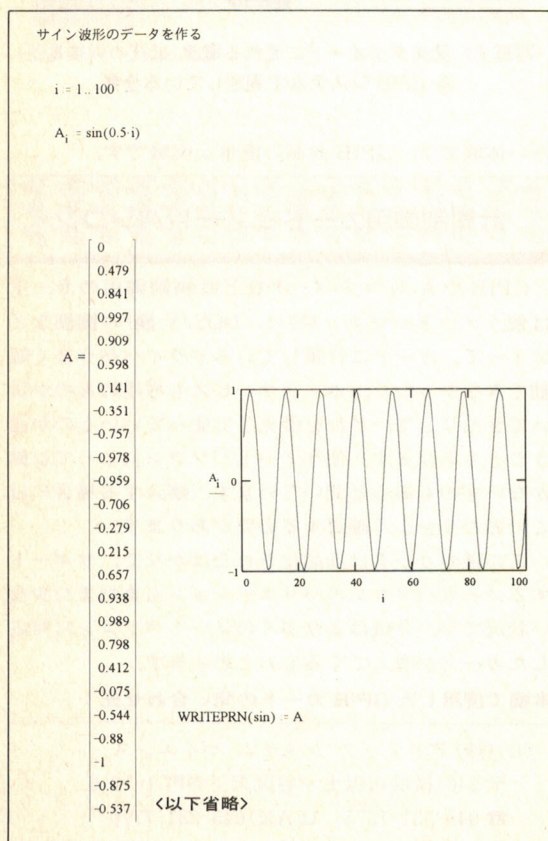
波形解析ソフトは，DOS/V対応のWindows環境で動作するものが多くあります．本誌で紹介した技術グラフ&データ解析ソフトウェアのOrigin(1995年5月号，6月号)や計測制御用グラフィックプログラミングLabVIEW，数学モデリングソフトウェアVisSim(1)などいろいろあります．もちろん数式処理ソフトとして知られているMathematica(1)やMathcad(1)にも波形解析に必要な機能がたくさん入っています．

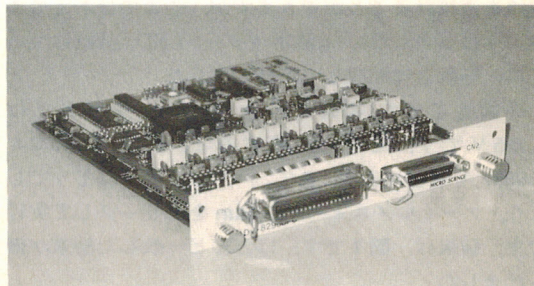
ここで紹介するマイクロサイエンス社の波形解析ソフトLaBDAQ(Ver.2.0)は，PC98上で動作する数少ない波形解析ソフトの一つです．LaBDAQはどんなことができるかというと，専用のA-Dコンバータボード(高速A-Dコンバータボード，**写真1**は同社のADM-8298BPC)から取り込んだデータの解析(波形解析)はもちろんのこと，別の実験で得られたデータ(ASCII形式)を取り込んで波形解析ができます．このソフトの波形解析の中心はなんといってもパワースペクトル解析です．

上述したDOS/V対応のソフトにはFFT解析はいわば標準で付いていますが，パワースペクトルになると限られます．LaBDAQはパワースペクトルを中心に，FFT解析や自己相関解析などができるソフトです．

さっそく，事例を用いてLaBDAQの使用例を紹介します．

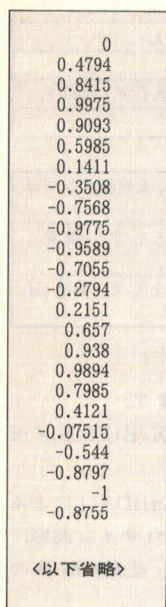
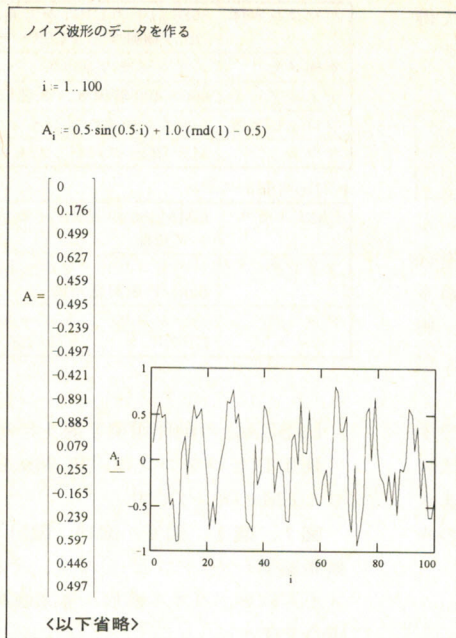
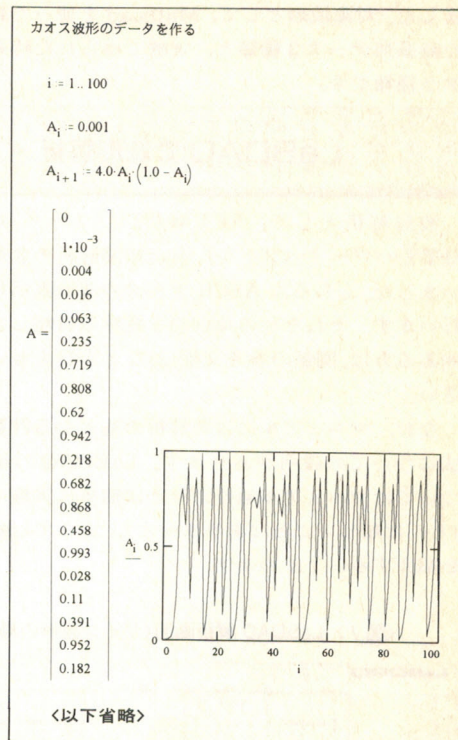
対象となるデータは(1)サイン波形，(2)ノイズ波形，(3)カオス波形，(4)電流波形A，(5)電流波形Bです．(1)，(2)，(3)の波形はMathcadを用いて計算で得たものです．(4)，(5)の波形は別の実験で得た過渡電流波形です．手元にあったものをそのまま使います．

〈図1〉 Mathcadのドキュメント画面(サイン波形)

## Mathcadで時系列データをつくる(1)

Mathcadには，計算で得たデータをフロッピにおとす機能があります．**図1**はサイン波形の時系列データ(ASCIIデータ)を生成するためのドキュメント画面です．Mathcadの使用法やドキュメント画面の詳細については省きますが［参考文献(1)を参照］，計算式は

〈写真1〉 高速A-Dコンバータボード(ADM-8298BPC)

〈図2〉サイン波形の時系列データ(ASCII データ)

```
0
0.4794
0.8415
0.9975
0.9093
0.5985
0.1411
-0.3508
-0.7568
-0.9775
-0.9589
-0.7055
-0.2794
0.2151
0.657
0.938
0.9894
0.7985
0.4121
-0.07515
-0.544
-0.8797
-1
-0.8755
```

<以下省略>

〈図3〉Mathcad のドキュメント画面(ノイズ波形)

〈図4〉Mathcad のドキュメント画面(カオス波形)

$A_i = \sin(0.5 * i)$ で与えます. i を1から100まで変化させたときのAの値を求めます. Aの値は一次元のマトリクスで表示されます. これが時系列データです.

この値をフロッピに **sin.prn** というファイル名でおとします. 関数 **WRITEPRN( )** を使って **WRITEPRN(sin)=A** とすれば, Mathcad が自動的に実行してくれます. **図2** は Windows 付属のメモ帳で読み取った **sin.prn** の ASCII のデータ(一次元のデータ列)です. なお, **図1** には, Mathcad のグラフ機能で ASCII データをグラフ化しています. サイン波形です.

〈図5〉ノイズ波形の時系列データ(ASCII データ)

```
0
-0.259
0.1141
0.5838
0.305
0.6221
-0.2553
0.0351
-0.5744
-0.8974
-0.8321
0.1357
-0.5206
-0.3835
0.3602
0.5708
0.1609
0.35
-0.2369
```

<以下省略>

〈図6〉カオス波形の時系列データ(ASCII データ)

```
0
0.001
0.003996
0.01592
0.06267
0.235
0.719
0.8081
0.6202
0.9422
0.2178
0.6816
0.8682
0.4578
0.9929
0.02823
0.1097
0.3908
0.9523
```

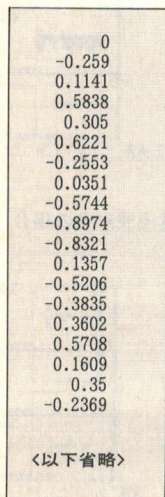
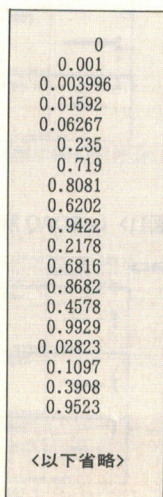
<以下省略>

同じようにして, ノイズ波形とカオス波形についても時系列データを生成します. それぞれの Mathcad ドキュメント画面は **図3** と **図4** です.

ノイズ波形の計算式はサイン波形 [0.5 * sin(0.5 * i)] にノイズとして乱数 [1.0 * (rnd(1) − 0.5)] を重畳させた $A_i = 0.5 * \sin(0.5 * i) + 1.0 * (\mathrm{rnd}(1) - 0.5)$ です. カオス波形の計算式はよく知られたロジスティックの式で, $f(t) = a * f(t-1) * (1 - f(t-1))$ の $a = 4.0$ とした場合です.

図中のグラフをみれば, ノイズ波形とカオス波形とは異なることがわかります. カオスは一見ランダムにみえていても実は規則性があります(ストレンジアトラクタという奇妙な形の軌跡を描く). **図5** と **図6** はそれぞれノイズ波形(**noise.prn**)とカオス波形(**chaos.prn**)の ASCII データです.

## 実験データ

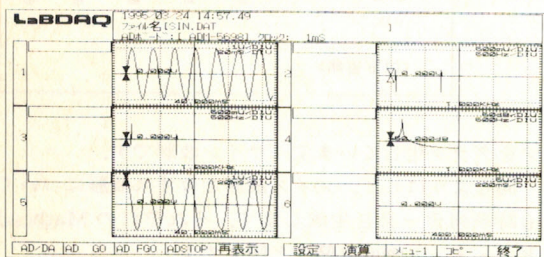
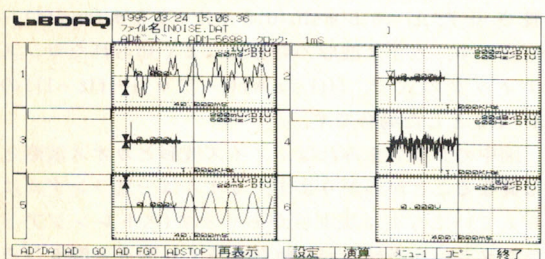
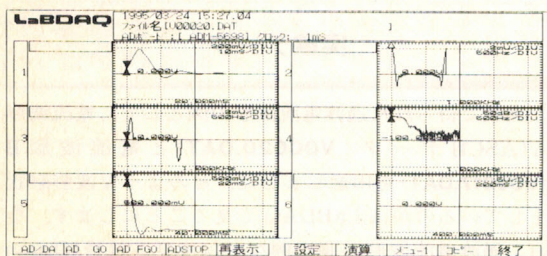
実験で得られた過渡電流波形の例として, 電流波形A(ASCII データ：**V00020.DAT**)と電流波形B(**V00017.DAT**)を用意しました. どのような波形形状をしているのかは LaBDAQ で見ることにします. なお, 過渡電流波形の測定例などは本誌5月号(Origin

を使った波形シミュレーション)を参照してください.

これでLaBDAQを使って波形解析する準備ができました. 対象波形として, Mathcadを使って計算で得た時系列データ3種類と, 実験で得られた時系列データ2種類です.

## LaBDAQで波形解析

波形解析として, FFT解析, パワースペクトル(FFTパワースペクトル), 自己相関解析の3種類を行いました. もちろんASCIIデータの波形表示(再現)も行います. それぞれの物理的な意味は省略します. 興味ある方は, 関係の参考文献(2)などをひもといてください.

なお, マニュアルには各解析の基となる計算式や手法についての説明があります. LaBDAQの使用法についてはマニュアル通りに行えば簡単に実施できます(とくに難しいことはありません). ここでは結果のみを紹介するにとどめます.

〈図7〉LaBDAQ解析画面(サイン波形の場合)

〈図8〉LaBDAQ解析画面(ノイズ波形の場合)

〈図10〉LaBDAQ解析画面(電流波形Aの場合)

〈表1〉LaBDAQの動作環境

▶必須システム

| | |
|---|---|
| パソコン本体 | 8086以上のCPUをもち, MS-DOS 3.1以上のOSが稼動するPC98パーソナルコンピュータ |
| 本体メモリ | メインメモリ640 Kバイト以上 |
| ディスプレイ | 600×400解像度, 8色以上 |
| フロッピ装置 | 3.5インチまたは5インチドライブ1台以上 |
| マウス | MS-DOSに対応したもの |

▶対応可能システム

| | |
|---|---|
| EMSメモリ | EMM386のEMSメモリ対応, 大容量サンプリング可能 |
| プリンタ | ハードコピー, レポート印刷として, PR系, ESC/P系対応可能 |
| プロッタ | ハードコピー, レポート印刷として, HP-GL, GP-GLコマンドに対応 |

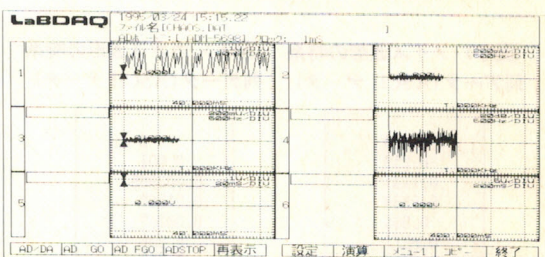
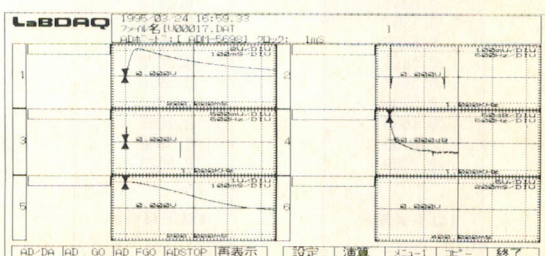
LaBDAQの動作環境を**表1**に示します.

使用したパソコンは, PC98note SX/E(EMSメモリ:2Mバイト)です.

**図7**, **図8**, **図9**, **図10**, **図11**はLaBDAQによる解析結果(ハードコピー)です. それぞれサイン波形, ノイズ波形, カオス波形, 電流波形A, 電流波形Bの場合です.

各図は六つの計測ウィンドウで構成されています. ウィンドウの数は1から16まで設定可能です. 計測ウィンドウには番号が割り当てられており, 順番にWNO1, WNO2, WNO3, WNO4, WNO5, WNO6となっています. 今回使用したのはWNO1~WNO5の五つのウィンドウです.

WNO1はASCIIデータの波形表示用です. WNO2はFFT解析の実数部表示用で, WNO3はFFT解析

〈図9〉LaBDAQ解析画面(カオス波形の場合)

〈図11〉LaBDAQ解析画面(電流波形Bの場合)

の虚数部表示用です．WNO4はパワースペクトル表示用でWNO2とWNO3のFFT解析から算出しています．WNO5は自己相関解析表示用です．

**(1) ASCIIデータから波形表示(ウィンドウWNO1)**

サイン波形，ノイズ波形，カオス波形いずれもMathcadで見たグラフとまったく同じです．当然の結果です．電流波形Aは少し振動性のある過渡電流波形でした．電流波形Bは“山形をした”過渡電流波形でした．実は，実験のときに，電流波形はA-Dコンバータを介してノートパソコンのモニターで観測していましたが，このとおりの波形でした．

**(2) FFT解析(ウィンドウWNO2，WNO3)**

サイン波形の場合は周期関数ですので，FFTの実数部，虚数部とも周期を表すスペクトルが得られています．ノイズ波形の場合もサイン波形をベースにしていますので，FFTには同様にスペクトルが見られます．波形だけを見るといかにも周期性のないランダムな波形に見えますが，FFT解析をしてみると周期性のあることがわかります．ただし，FFTには少しノイズ成分(雑音レベル)が加わっています．

一方，カオス波形の場合には，その性質上，FFTにはスペクトルは認められません．雑音レベルのみです．

電流波形AにはFFTのスペクトルが見られます．少し幅のあるスペクトルです．スペクトルに幅があるということは，いろいろな周波数成分の波形が含まれているということになります．電流波形Bはスペクトルはありません(一見，ありそうに見えますがサイン波形と比べてみるとないことがわかる)．

**(3) パワースペクトル(ウィンドウWNO4)**

サイン波形，ノイズ波形については顕著なスペクトルがみられます．ただし，ノイズ波形の場合は雑音レベルが相当加わっています．カオス波形は雑音レベルだけでスペクトルは見られません．電流波形Aは幅をもったスペクトルが認められます．電流波形Bはスペクトルは認められません．

**(4) 自己相関解析(ウィンドウWNO5)**

自己相関解析とは一言でいうと，現在の自分自身が過去の履歴にどの程度影響されているか，時間的にどの程度相関があるのかをみる解析手段です．

サイン波形の場合はいつまでたってもサイン波形ですから，波形が続くかぎり相関があるということです．ノイズ波形も波形のベースはサイン波形ですから，同様に相関があります．いずれも自己相関はサイン波形です．

カオス波形はどうでしょう．もともと“混沌”としていて相関なんてありませんから，ウィンドウWNO5で見るようになんら相関波形は認められません．

電流波形Aは最初に周期的な振動をしてから次第に0に落ち着いていく波形です．したがって，最初は相関があり周期的な振動をしていますが，次第に相関がなくなっていきます．電流波形Bもほぼ同様の傾向です．いずれにしても時間経過とともに次第に相関がなくなっていく様子がみられます．

LaBDAQにはこの他にも，直流成分抽出，直流成分除去，単純移動平均，2次多項式適合平滑，FFT位相スペクトル，FFTクロススペクトル，相互相関といった解析機能があります．

このように，PC98環境の解析ソフトとしては非常に興味ある解析機能がついています．PC98のパソコン資産を活用して計測や計測データの波形解析する場合には大変有用なツールになりうると思います．機会がありましたならぜひ試してみてください．

最後に，同社の関係の製品を紹介します．

① 高速A-D変換ボード，D-A変換ボード，前置モード

| | | |
|---|---|---|
| A-D変換ボード | ADM-5298BPC<br>ADM-5598BPC<br>ADM-5698BPC<br>ADM-5898BPC<br>ADM-1498BPC<br>ADM-1598BPC | ADM-1698BPC<br>ADM-1998BPC<br>MFU98-401B<br>ADM-5498BPC<br>ADM-5198BPC |
| D-A変換ボード | MDA-2898BPC<br>(12 bit/2 CH 絶縁高速2 K FIFO付き) | |
| 前置ボード | PFA-6598BPC<br>(ソフトプログラマブル・ユニバーサルフィルタ＆アンプ) | |

② 高速A-Dサンプリング・波形表示ソフト
EGLaB for Windows

③ データ処理ライブラリ
▶LabSYS-98
▶LaBDAQの言語版(CまたはBASICで使用可能な波形解析用の関数セット)

④ フィルタ設計
ADFILTER-DII

⑤ WINDOWSプログラム開発キット
WINKIT-(ボード番号)-KIT

**問い合わせ先**：マイクロサイエンス(株)
〒167 東京都杉並区西荻北3丁目2番3号
☎ 03-3396-8362　FAX：03-3301-5593

**◈ 参考文献 ◈**

(1) 臼田・井上・伊藤；パソコンによる数式処理ソフト活用法，CQ出版(株)．
(2) 浜田望；よくわかる信号処理，オーム社．

# 第5章

## インテリジェントリモートI/Oを使ったデータ収集システム

# RS-232Cによるリモート電圧/電流測定の方法

臼田昭司

## リモートI/Oモジュールとは

リモートI/Oモジュールとは，LANのメーカーで知られているコンテック社[1]が商品化しているデータ収集のためのモジュールです．

マイクロプロセッサを内蔵し，プラスチックケースにコンパクトに実装されており(**写真1**)，手のひらに乗ります．

パソコン(ホストコンピュータ)の通信ポートであるRS-232Cから簡単なコマンドをモジュールに送り，このコマンドを受けてセンサーとしてデータ(電圧，電流，温度など)を収集(計測)することができます．

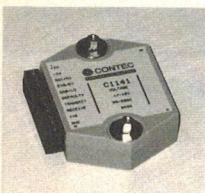
モジュールがパソコンから離れた場所にあれば，離れた場所での計測を手元のパソコンでリモートできます(**図1**)．

モジュールには**表1**のようなバリエーションが用意されています．ここでは，電圧型アナログ入力C1141(以下，電圧モジュール)と電流型アナログ入力C1251(以下，電流モジュール)の2種類のモジュールについて試用例を紹介します．

## リモートI/Oモジュールを使うための準備

**(1) ハードの準備**

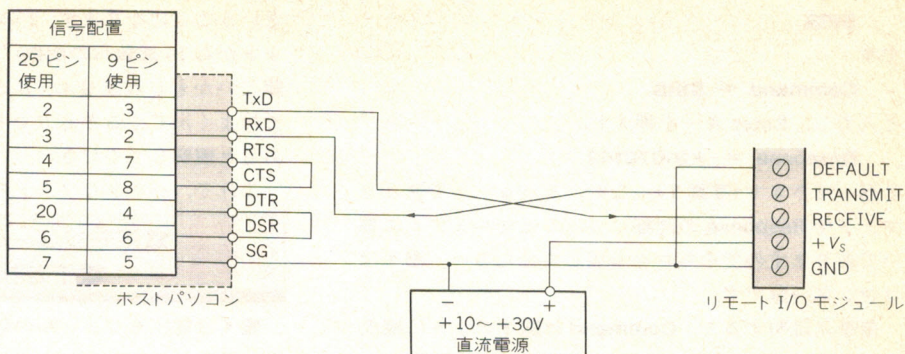
**図2**はパソコンのRS-232C(9ピンと25ピンの場合)とモジュールの接続関係を示したものです．

モジュールを駆動するための電源については，モジ

**〈写真1〉リモートI/Oモジュール**(電圧型アナログ入力C1141)

**〈図1〉▶ 各種モジュールを使って離れた場所での分散制御やデータ収集**

**〈表1〉モジュールのバリエーション**

| | |
|---|---|
| C1000シリーズモジュール | ▶C1100シリーズ…電圧型アナログ入力：C1141/C1142<br>▶C1200シリーズ…電流型アナログ入力：C1251/C1252<br>▶C1300シリーズ…J型熱電対センサ入力：C1311/C1312<br>K型熱電対センサ入力：C1321/C1322<br>▶C1400シリーズ…RTD入力：C1411/C1412<br>▶C1500シリーズ…ブリッジ入力：C1521/C1522<br>▶C1600シリーズ…パルス入力：C1601/C1602<br>▶C1700シリーズ…固定割り付けディジタル信号入出力：C1701/C1702<br>任意割り付けディジタル信号入出力：C1711/C1712 |
| C3000シリーズモジュール | ▶電圧型アナログ出力：C3141/C3142 |

〈図2〉
パソコンのRS-232Cとモジュール，および電源との基本接続関係
(センサ端子+IN，-INはまだ接続していない)

〈表2〉各言語に対応したサンプルプログラム

| 言　語 | | ファイル名 | |
|---|---|---|---|
| | | パソコン本体ポート使用 | 拡張ボード使用 |
| MS-C | Ver. 5.1以上 | IRIOAC01.C | IRIOAC02.C |
| Borland C++ | Ver. 2.0以上 | IRIOBC01.C | IRIOBC02.C |
| QuickBASIC | Ver. 4.5以上 | IRIOQB01.BAS | IRIOQB02.BAS |
| GW-BASIC | Ver. 2.0以上 | IRIOGW01.BAS | IRIOGW02.BAS |

＊拡張ポート使用とはコンテック社の製品を使用した場合

〈写真2〉
英語版QBASICはC：\DOS\QBASIC ⏎で起動

ュール内にスイッチングレギュレータ(安定化回路)を内蔵しているため，DC 10 V～30 Vの広い範囲のものが使用できます．なお，接続に際しては耐ノイズ性を考えて，ツイストペア・ケーブル(平衡線)を使用します．

RS-232Cのピン番号を間違わないように注意深く接続します．この接続は電圧モジュールも電流モジュールも同じです．モジュールの電源としては手持ちのDC 12 V電源(TDK製FMP12-R25)を使いました．ホストパソコンについては，DOS 6.2/VとWindows 3.1がプリインストールされたDOS/Vパソコン(AST製BRABO LC 4/33，9ピンRS-232C)を使いました．

**(2) ソフトの準備**

パソコンからモジュールにコマンドを送るためには専用の通信プログラムが必要ですが，各言語に対応したサンプルプログラム(**表2**)がメーカーには用意されています．

実験では，DOS 6.2 Vの中にある英語版QBASIC(**C:\DOS\QBASIC.＊**)を使いました(**写真2**)．このQBASICはインタプリタで，マウスも使用できます．英語モードとはいっても BASICに慣れた方なら簡単に使えます．

さっそく，QBASICを立ち上げ，サンプルプログラム(**IRIOQB01.BAS**)を読み込み(**写真3**)，プログラムを実行します(**写真4**)．

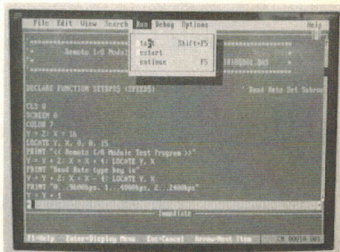
上記(1)の接続がまちがっていなければ，**<<Remoto I/O Module Test Program>>** と表示され，通信速度(ボーレート)はいくらか(**Baud Rate Type Key in**)と聞いてきます．モジュールのデフォルトは300 bpsに設定されていますので，番号で**5**と入力します．するとモジュールは，何か命令(コマンド)を入力してくださいという意味の，

**Command** ＝ □

を返します．モジュールが応答しました．

それではモジュールの通信条件など，デフォルト値を確認するために，コマンドを入力してみます．現在の設定データを確認するためのコマンドは，

◀〈写真3〉
QBASICでサンプルプログラム(IRIOQB01.BAS)を読み込む

〈写真4〉▶
サンプルプログラムを実行(start)する

**$1RS**

です．

**Command ＝ $1RS**

と入力し，Enter キーを押すと，

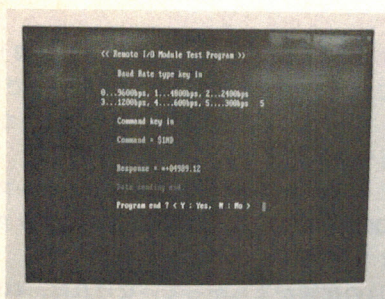
**Response ＝ ＊31070142**

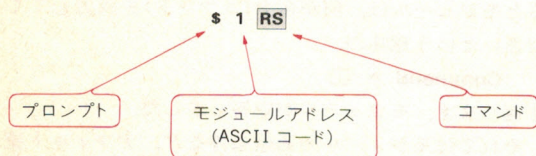
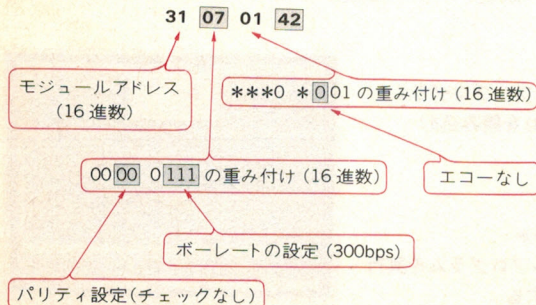
と返ってきます(**写真 5**)．なお，モジュールで使うコマンドや **Response** の詳細についてはマニュアルに書いてありますのでそちらを参照してください．**表 3** はコマンド一覧です．

簡単に説明すると，**Command** は**図 3** のように構成されます．プロンプト＄とモジュールアドレス 1 はコマンドの前に必ず付けます．“ひらけごま！”といった意味のおまじないと思ってください．

Response については，**図 4** のようになります．

これらをまとめると，モジュールのデフォルトは，

▶ボーレート　　：300 bps
▶パリティ　　　：なし
▶データビット　：8 ビット
▶ストップビット：1 ビット

〈**写真 5**〉Command＝$1RS を入力した後，Response＝＊3107142 が返ってきたところ

〈**図 3**〉Command のフォーマット

〈**図 4**〉Response のフォーマット

ということになります．上の説明では，“重み付け(8 ビットから 16 進数への変換)”などがわからないと少し難しいかもしれません．8 ビットマイコンである Z80 ではよく出てくる考え方です．大略このようなものであると理解してください．

これで，ハードとソフトの準備が完了しました．

## 電圧モジュール(C1141)で電圧を計測する

**表 4** は電圧モジュールの仕様です．変換精度や直線性，温度ドリフトなどから，かなりの高精度な測定が期待できます．

〈**表 3**〉モジュールで使うコマンド一覧

● ユーザーコマンド

| コマンド | | 機　能 |
|---|---|---|
| **ACK** | Acknowledge | 通信ハンドシェーク |
| **DI** | Read Alarms Digital Input | ディジタル入力，アラームステータス読み出し |
| **DO** | Set Digital Output | ディジタル出力 |
| **ND** | New Data | センサーデータ読み出し |
| **RD** | Read Data | センサーデータ読み出し |
| **RE** | Read Event Counter | イベント数の読み出し |
| **RH** | Read High Alarm | 高アラーム値とタイプの読み出し |
| **RL** | Read Low Alarm | 低アラーム値とタイプの読み出し |
| **RS** | Read Setup | 設定データの読み出し |
| **RZ** | Read Zero | 出力オフセットレジスタの読み出し |
| **WE** | Write Enable | 書き込み禁止状態の解除 |

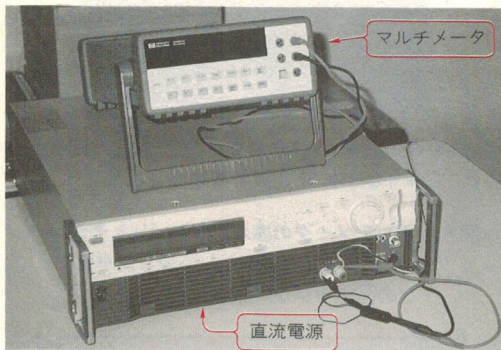
● 書き込み禁止コマンド

| コマンド | | 機　能 |
|---|---|---|
| **CE** | Clear Events | イベントカウンタを 0 にクリアする |
| **CZ** | Clear Zero | 出力オフセットレジスタを 0 にクリアする |
| **DA** | Disable Alarms | アラーム機能を禁止する |
| **EA** | Enable Alarms | アラーム機能を許可する |
| **HI** | High Alarm Limit | 高アラームの値とタイプの設定 |
| **LO** | Low Alarm Limit | 低アラームの値とタイプの設定 |
| **CA** | Clear Alarms | 高，低アラームの両方を OFF にする |
| **RR** | Remote Reset | プログラムリセット |
| **SU** | Setup Module | モジュールのセットアップ |
| **SP** | Set Setpoint | セットポイントの設定 |
| **TS** | Trim Span | モジュールの正確性を調整する |
| **TZ** | Trim Zero | 出力データのオフセットをヌルにし，偏差値出力を作成 |

〈**表 4**〉電圧モジュール(C1141)の仕様

| 項　目 | 内　容 |
|---|---|
| 入力電圧レンジ | −10 V〜＋10 V |
| 変換精度 | ±4 LSB(0.01%のフルスケールレンジ) |
| リニアリティエラー | 0.02%のフルスケールレンジ(max) |
| ゼロドリフト | ±1 カウント max(オートゼロ) |
| 温度ドリフト | ±50 ppm/°C(max) |
| 入力インピーダンス | 1 MΩ(min) |
| ディジタル入力 | 1 点(イベントカウンタとして使用可能) |
| ディジタル出力 | 2 点 |

電圧源として，菊水電子工業製[2]の直流電源(PAX35-10)を，電源電圧のチェック用にYHP社[3]のマルチメータ(HP34401A)を使いました(**写真6**).

それぞれの簡単な仕様を**表5**と**表6**に示します.

**〈写真6〉電圧源として直流電源(下)と，電源電圧のチェック用にマルチメータ(上)を使用**

**〈表5〉マルチメータ HP34401A の一般仕様**

| 一般仕様 | |
|---|---|
| 電源 | 100 V/120 V/220 V/240 V　±10% |
| 電源周波数 | 45～66 Hz，360 Hz～400 Hz<br>電源投入時に自動的に感知される |
| 消費電力 | 最大25 VA(平均10 W) |
| 動作環境 | 0～55℃でフル確度<br>30℃で80%R.H.までフル確度 |
| 保管環境 | −40～＋70℃ |
| ラック寸法 | 88.5 mm(高さ)×212.6 mm(幅)×348.3 mm(奥行) |
| 重量 | 3.6 kg |
| 安全規格 | CSA，UL-1244，IEC-1010に適合 |
| EMI | MIL-461C，FTZ1046，FCC |
| 振動と衝撃 | MIL-T-28800E type III，Class 5 |
| 補償 | 3年間 |
| **付属アクセサリ** | |
| プローブ(1)，ワニロ(1)，ピンタイプ(2)・アタッチメント付きテストリード・キット.<br>ユーザーズガイド，サービスガイド，テストレポート，電源コード | |
| **トリガとメモリ** | |
| 読み取りホールド感度 | 読み取り値の0.01%，0.1%，1%，10% |
| 1トリガ当たりのサンプル数 | 1～50,000 |
| トリガ遅延 | 0～3600秒<br>(ステップサイズ10 μs) |
| 外部トリガ遅延 | <1 ms |
| 外部トリガジッタ | <500 μs |
| メモリ | 512読み取り値 |
| **演算機能** | |
| ヌル，Min/Max/Avg，dB，dBm，リミットテスト(TTL出力付き)<br>dBm基準抵抗：50，75，93，110，124，125，135，150，250，300，500，600，800，900，1000，1200，8000 Ω | |
| **標準プログラミング言語** | |
| ▶ SCPI(プログラマブル計測器用標準コマンド)<br>▶ HP 3478A言語エミュレーション<br>▶ Fluke 8840A，Fluke 8842A言語エミュレーション | |
| **リモートインターフェース** | |
| HP-IB(IEEE-488.1，IEEE-488.2)およびRS-232C | |

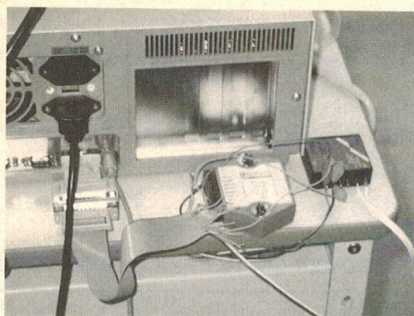
直流電源の＋－出力端子(OUTPUT)と電圧モジュールのセンサー端子(＋IN，－IN)を接続します(**写真7**).

モジュールでセンシングした電圧を読むためのコマンドは，

**〈表6〉DC POWER SUPPLY PAX35-10の電気的仕様**

| 項目 | | | | PAX35-10 | 単位 |
|---|---|---|---|---|---|
| 入力特性 | 定格入力電圧 | | | AC 100 V±10%，50/60 Hz | |
| | 入力電流(AC 100 V，FULL LOAD) | | | 8.5 A | |
| | 突入電流 | | | 13A〔at V(in)＝AC 110 V〕 | |
| 出力設定 | 電圧設定 | 設定範囲 | | 0～35.00 | V |
| | | 分解能 | | 10 | mV |
| | 出力電圧 | 出力の誤差 | | ±40 | mV typ |
| | | 温度係数 | | 100(35 typ) | ppm/℃ |
| | 電流設定 | 設定範囲 | | 0～10.00 | A |
| | | 分解能 | | 10 | mA |
| | 出力電流 | 出力の誤差 | | ±40 | mA typ |
| | | 温度係数 | | 150(50 typ) | ppm/℃ |
| 表示 | ディジタルメータ | 出力電圧 | 表示の誤差 | 0.07%＋10 | mV typ |
| | | | 温度係数 | 100 | ppm/℃ typ |
| | | 出力電流 | 表示の誤差 | 0.3%＋30 | mA typ |
| | | | 温度係数 | 150 | ppm/℃ typ |

| 項目 | | | | PAX35-10 | 単位 |
|---|---|---|---|---|---|
| 定電流特性 | ノーマルモード | リプル | RMS | 2 | mA(RMS) |
| | | 負荷変動 | | 7 | mA |
| | | 電源変動 | | 1 | mA |
| | | 立ち上がり | | 50 | ms typ |
| | | 立ち上がり | | 50 | ms typ |
| | ファーストモード | リプル | RMS | 3 | mA(RMS) |
| | | 負荷変動 | | 10 | mA |
| | | 電源変動 | | 1 | mA |
| | | 立ち上がり | 50 μs | 50 | μs(typ) |
| | | | 500 μs | 500 | μs(typ) |
| | | | 5 ms | 5 | ms(typ) |
| | | 立ち下がり | 50 μs | 50 | μs(typ) |
| | | | 500 μs | 500 | μs(typ) |
| | | | 5 ms | 5 | ms(typ) |

| 項目 | | | | PAX35-10 | 単位 |
|---|---|---|---|---|---|
| 定電圧特性 | ノーマルモード | リプル | RMS | 0.2 | mV(RMS) |
| | | | P-P | 3 | mV(P-P)typ |
| | | 負荷変動 | | 1 | mV |
| | | 電源変動 | | 1 | mV |
| | | 過渡応答 | | 50 | μs(typ) |
| | | 立ち上がり | | 50 | ms(typ) |
| | | 立ち下がり | | 50 | ms(typ) |
| | ファーストモード | リプル | RMS | 2 | mV(RMS) |
| | | | P-P | 10 | mV(P-P)typ |
| | | 負荷変動 | | 1 | mV |
| | | 電源変動 | | 1 | mV |
| | | 過渡応答 | | 100 | μs(typ) |
| | | 立ち上がり | 50 μs | 50 | μs(typ) |
| | | | 500 μs | 500 | μs(typ) |
| | | | 5 ms | 5 | ms(typ) |
| | | 立ち下がり | 50 μs | 50 | μs(typ) |
| | | | 500 μs | 500 | μs(typ) |
| | | | 5 ms | 5 | ms(typ) |

〈写真 7〉 直流電源の＋－出力端子(OUTPUT)と電圧モジュールのセンサ端子(＋IN, －IN)などを接続(RS-232C やモジュールの電源など基本接続はすんでいる)

〈写真 8〉 Command＝$1ND と入力した後, Response＝＊＋04989.12 と返ってきたところ

〈表 7〉
マルチメータの表示(単位：V)とモジュールの ND 値の比較

| マルチメータ | モジュール |
|---|---|
| 0.9908 | ＊＋00990.72 |
| 1.9929 | ＊＋01992.64 |
| 2.9957 | ＊＋02995.20 |
| 3.9891 | ＊＋03988.80 |
| 4.9891 | ＊＋04989.12 |
| 5.9912 | ＊＋05991.36 |

〈図 5〉 マルチメータの表示(単位：V)とモジュールの ND 値(電圧値として読む, 単位：V)との関係をグラフ化

**Command ＝ $1ND**

です. ND は "New Data" の意味です.

● **測定**

直流電源の電圧を 5 V に設定しました. マルチメータの表示は 4.9891 V です. 次に, パソコン側でコマンド **$1ND** を入力します. すると,

**Response ＝ ＊＋04989.12**

と返ってきました(**写真 8**). この表示(以下, **ND** 値)は 4.98912 V という意味ですが, マルチメータの読みと一致しました.

**表 7** と **図 5** は 1 V から 6 V までの測定例です. マルチメータの表示とモジュールの **ND** 値はほとんど一致しました. なお, **Response** の表示桁はデフォルト(**＋＊＊＊＊＊.00**)から**＋＊＊＊＊＊.＊＊**に設定し直した場合です(設定方法などはマニュアルを参照).

〈表 8〉 電流モジュール(C1251)の仕様

| 項　目 | 内　容 |
|---|---|
| 入力電流レンジ | ＋4 mA～＋20 mA |
| 変換精度 | ±16 LSB(0.04%のフルスケールレンジ) |
| リニアリティエラー | 0.04%のフルスケールレンジ(max) |
| ゼロドリフト | ±1 カウント max(オートゼロ) |
| 温度ドリフト | ±50 ppm/℃(max) |
| 電圧ドロップ | ±0.1 V(max) |
| ディジタル入力 | 1 点(イベントカウンタとして使用可能) |
| ディジタル出力 | 2 点 |

## 電流モジュール(C1251)で電流を計測する

**表 8** は電流モジュールの仕様です. 測定できる電流範囲は＋4 mA～＋20 mA となっています. この範囲では, 変換精度や直線性, 温度ドリフトなどは電圧モジュールと同様に高い精度をもっています.

菊水電子工業の直流電源(OUTPUT)に 500 Ω のホーロー抵抗を接続して定電流回路を作りました(**図 6**, **写真 9**). 電流値は直接マルチメータで測定(チェック)します. モジュールのセンサー端子の＋IN には電源の＋側を, －IN には電源の－側をそれぞれ接続します(＋と－を逆に接続しないように注意が必要).

● **測定**

電流を 4 mA に設定しました. マルチメータの読みは 3.8491 mA でした. これを電流モジュールでセンシングします.

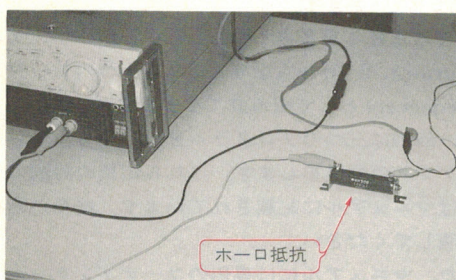
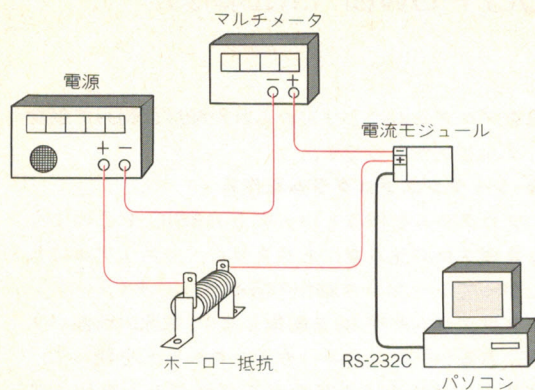
**Command ＝ $1ND**

と入力します. **ND** 値は,

**Response ＝ ＊＋00003.85**

と返します. 3.85 mA の意味ですが, ぴたりと一致しました.

このようにして, 19 mA まで測定した例が**表 9** と**図 7** です. 正確なものです. 直線性が保たれています.

〈図6〉電流モジュール電流をセンシングするための定電流回路

〈写真9〉ホーロー抵抗などを使って定電流回路を作る

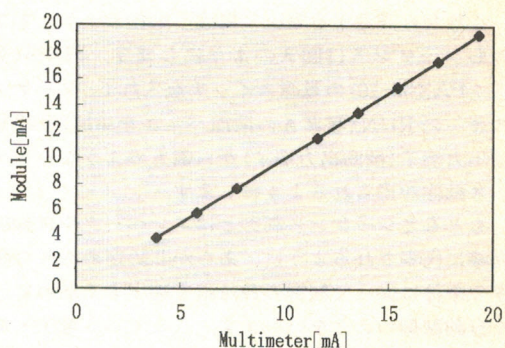
〈表9〉マルチメータの表示(単位：mA)とモジュールのND値との比較

| マルチメータ | モジュール |
|---|---|
| 3.8491 | *+00003.85 |
| 5.7857 | *+00005.76 |
| 7.7043 | *+00007.70 |
| 11.5689 | *+00011.57 |
| 13.504 | *+00013.50 |
| 15.424 | *+00015.42 |
| 17.36 | *+00017.36 |
| 19.3 | *+00019.36 |

〈図7〉マルチメータの表示(単位：mA)とモジュールのND値(電流値として読む，単位：mA)との関係をグラフ化

## 計測・制御にバリエーションも豊富です

ここで紹介したモジュールは電圧型アナログ入力と電流型アナログ入力の場合です．実は，その他にも熱電対センサ入力やブリッジ入力など計測・制御用に特化したバリエーションが用意されています(**表1参照**)．

上記試用では，モジュールはパソコン(RS-232C)の近くに置きましたが，実際の応用に際しては離れた場所にあるモジュールをパソコンでリモートしながら計測や制御をするといった使い方がこのモジュール本来の使用法です．

もちろん，実験室などで電圧，電流，温度といったデータを手軽かつ正確にモニター(計測)する場合にも有効と思います．いろいろな利用法が考えられます．

このモジュールは新しい製品指向のA-Dコンバータの一つといえるでしょう．

(1) 商品名：REMOTE I/O MODULE
㈱コンテック　☎03-5443-2135
(2) 商品名：PROGRAMMABLE DC POWER SUPPLY　型式 PAX35-10
菊水電子工業㈱　☎045-475-1171
(3) 商品名：マルチメータ　型式 HP34401A
日本ヒューレット・パッカード㈱
☎03-3335-8177

## コラム●直流電源(Programmable DC Power Supply)でシーケンス動作をプログラム

### ● シーケンス動作とは

菊水電子工業製のProgrammable DC Power Supply(工業用直流安定化電源，**写真A**，以下PDPSと略す)PAXシリーズは単なる安定化電源ではなくて，所望の電圧・電流動作モード(シーケンス動作)をあらかじめプログラムにより設定することができます．

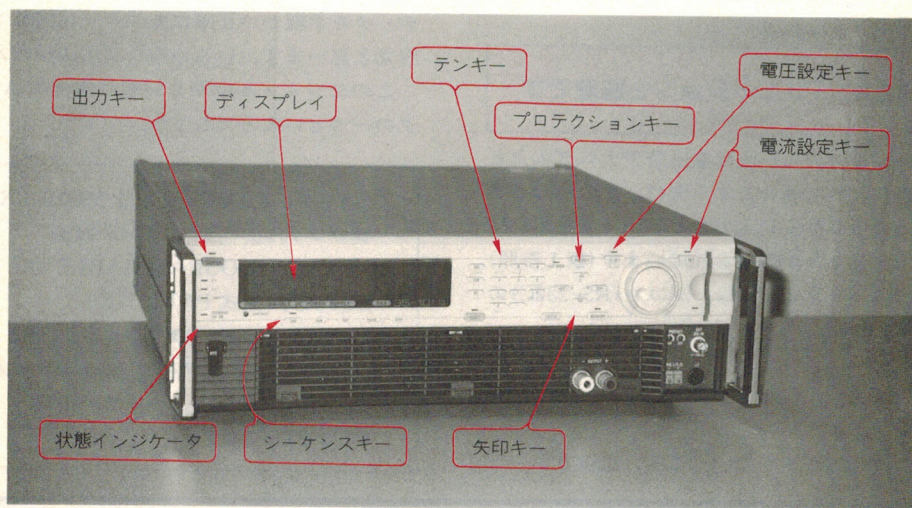
これから設定する電圧と電流の動作モード，すなわちシーケンスは**図A**のようにします．PDPS(型式：PAX35-10)の電源スイッチを入れて，シーケンスキーのRUN(**写真A**の前面パネルを参照)を押せば出力端子(補助出力端子)から**図A**のようなシーケンス動作が得られるようにします．

もともとシーケンス制御とはエレベータや自動販売機に代表されるように，あらかじめ定められた順序や条件に従って制御の各段階を順序よく進めていく自動制御のことをいいます．ここでいう電圧・電流のシーケンス動作もまったく同じです．電圧や電流をプログラムにより定められた順序に従って制御しようというものです．

### ● シーケンスプログラムを作る

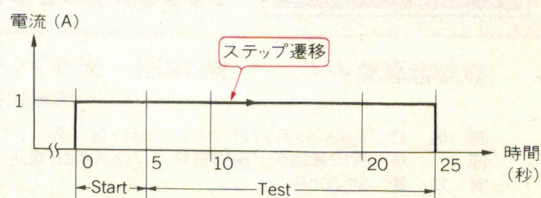
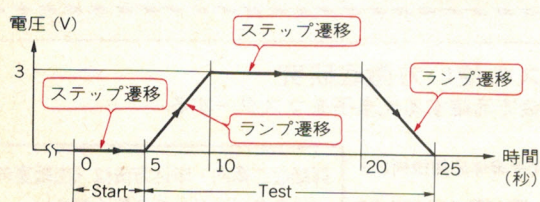
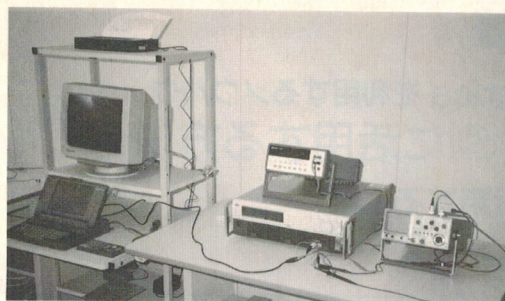
プログラムを作るといってもBASICやCのような言語プログラムではありません．マニュアルにしたがってシーケンス動作の各パラメータ(シーケンスプログラムと呼ぶ)を編集します．電源の前面パネルにあるシーケンスキーやテンキーなどを使って，液晶ディスプレイを見ながらプログラムを作って(編集して)いきます．

**図A**の電圧・電流モードは，三つのプログラム(Program1，2，3)からなり，これを一つのシーケンス(Sequence)として組み立てていきます．手順としては，最初に三つのプログラムを編集して，続いてシーケンスの編集をします．これら編集の詳細についてはマニュアルに記載されています．機会があれば参照してください．

また，マニュアルには専用のコーディングシート

〈写真A〉 Programmable DC Power Supply(PAX35-10)

〈図A〉 電圧・電流動作モード(シーケンス動作)の例

〈写真B〉シーケンス動作測定全景

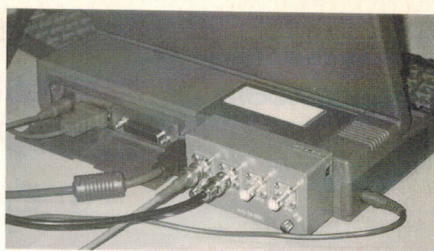
〈写真C〉A-Dコンバータとノートパソコン

(シーケンス作成用シート)が付録として付いています．先に，このシートにプログラムとシーケンスを書いてからキー入力をするようにします．

### ● シーケンス動作を見る

**図A**のシーケンス動作を波形表示させれば動作の理解などがわかりやすいと思います．電源の出力端子には抵抗などの負荷をつなげば最大1Aの電流を流すことができますが，無負荷状態で電圧の変化(電圧波形)のみを見ることにします．動作時間は秒オーダですので，オシロスコープを遅い掃引時間(例えば0.2 ms/div)にして目で電圧の変化の様子を見てみます．

それではシーケンス動作を実行させます(**写真B**)．オシロスコープのブラウン管を見つつ掃引のタイミングを図り，電源の前面パネルのRUNキーを押します．…最初に，電圧がランプ状に上昇していく様子が見られました．電圧が上昇しきったところでマルチメータを見たら約3Vを指示しています．数秒経過すると今度はランプ状に減少していきます．マルチメータの指示は0Vです．

これで，**図A**のシーケンス動作がおおむね確認できました．

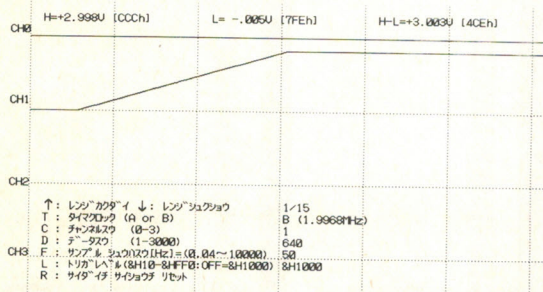
それでは，いまの電圧変化を電圧波形としてプリントアウト(ハードコピー)してみます．これには電子計測が一番です．手元にコンパクトで大変手軽なノートパソコン対応のA-Dコンバータ[(1)]がありました(**写真C**)．

BASICで作成した波形表示プログラムをDOS/BASICとともにノートパソコン(98NOTE SX/E)にインストールして測定開始です．

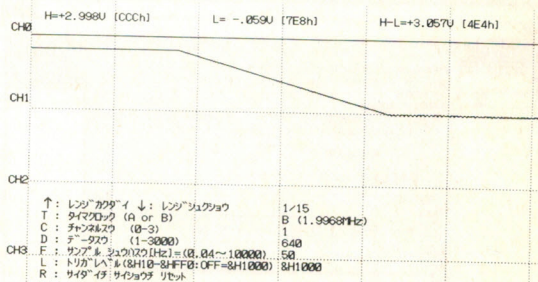
**図B**と**図C**はそれぞれ電圧の上昇過程と減少過程の測定例です(図中のメッセージなど詳細は省略)．シーケンスプログラムで設定したとおり，電圧はランプ状に遷移しています．予定どおりのシーケンス動作です．ランプ遷移の最大電圧も設定値2.998 Vとなっています．

### ● 他にもいろいろな機能がある

菊水電子工業のPDPSは，ここで紹介した機能のほかにもいろいろな応用操作があります．GPIBを使ったリモートセンシングもその一つです．“電源”と一口に言っても安定化機能だけでなく，パソコン計測に対してはフレキシビリティに対応が可能になってきています．“重い”電源というイメージが少しずつ解消できそうです．

(1) 商品名：A-Dコンバータユニット
…A/DO4-98n，
インターフェースユニット…I/F-98n
山崎電気㈱ 茨城県稲敷郡阿見町阿見5060-7
☎ 0298-87-4488

〈図B〉ランプ状に上昇する電圧波形

〈図C〉ランプ状に減少する電圧波形

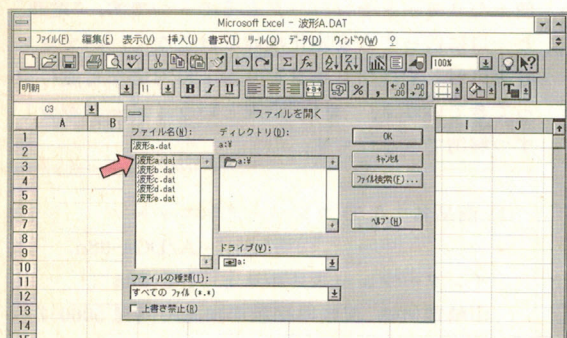
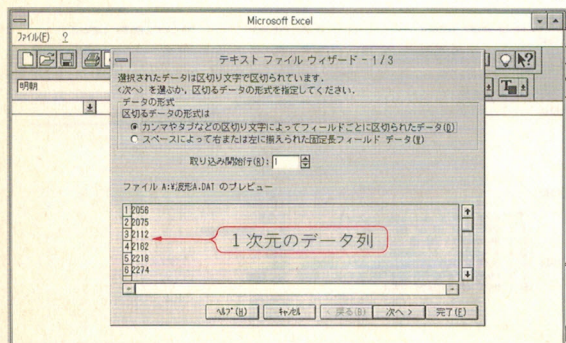
# 第6章

## Excel, Lotus1-2-3, Mathematica を利用するノウハウ
# 汎用パソコンソフトを計測に活用する方法
## ① Excel を活用する

臼田昭司

計測などで得たデータは，グラフにすると現象を理解する一助となります．ここで，得られたグラフはどのような傾向になっているのか，あるいはどのような曲線の式で近似できるのか，といった波形解析が必要になることがあります．

ここでは，これらについて事例を紹介します．対象とする計測波形は過渡電流波形です．この波形がどのような形状をしているのかをはじめに見ます．次に，波形がどのような近似式でフィット(Fit)できるのかを調べます．

解析用のツールとして，いずれも表計算ソフトの定番である Excel 5.0 を使います．

〈図1〉フロッピの中の ASCII データを開く

〈図2〉テキストファイル・ウィザードでデータ形式を見る

### ● データの読み込み

過渡電流波形のデータは，ASCII 形式の時系列データです．このデータが入っているフロッピを DOS/V パソコンの A ドライブに入れて，Excel のメニューから「ファイル(F)を開く」を選び，A ドライブのファイルの中身を見ます(**図1**)．**波形 a.dat**, …が過渡電流波形の各種データです．

このままでは，いずれもどのような波形形状をしているのかはわかりません．この中から **波形 a.dat** を選びます．**図2** のようなテキストファイルをどのような形式で読み込むのかを聞いてきます．図中のプレビュに表示されている1次元のデータ列のままでよいので，このまま **完了(F)** をクリックします．**図3** を見てください．1次元のデータ列が列 A に表示されました．

実は，この時系列データは100個ありますので，画面には見えていませんがセル A1 から A100 までデータで埋まっています．次に，この時系列データを実際の電流値に換算します．換算式は測定条件から，

**B1=(A1-2048)＊0.0024414/2.2** [A]

〈図3〉列 A に取り込まれた1次元のデータ列(ASCII データ)

〈図4〉セル B1 に ASCII データを電流値に換算する式を入れる

です．この式はとくに意味があるわけではありません．筆者の実験系で得られたものです．この換算式を使って，セルA1のデータを実際の電流値に検算してセルB1に入れます．セルB1に換算式を代入します(**図4**)．この式をB2からB640までコピーします(**図5**)．

**波形 a.dat** はどのような波形形状をしているのでしょうか？

さっそくExcelのグラフ機能を使ってグラフにしてみましょう．グラフウィザードを使ってグラフの書式などを指定します(**図6**)．**波形 a.dat** の波形形状は振動を伴う電流波形でした(**図7**)．

このように，グラフにすると現象が理解できます．この電流波形は $R$-$L$-$C$ 直流回路でスイッチをONにした瞬間の電流波形を計測したもので，いわゆる過渡現象をみたものです．

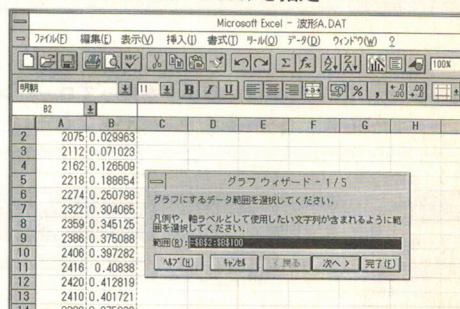
## ● 波形のフィッティング

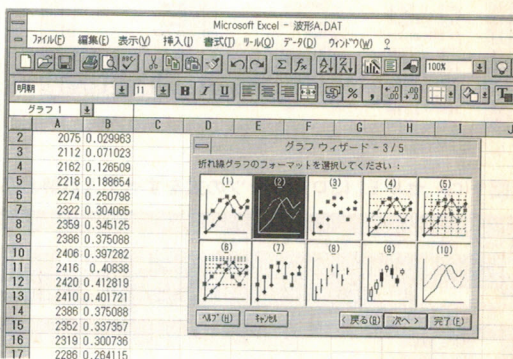
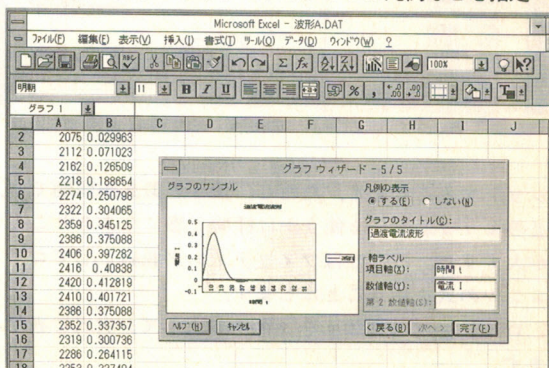
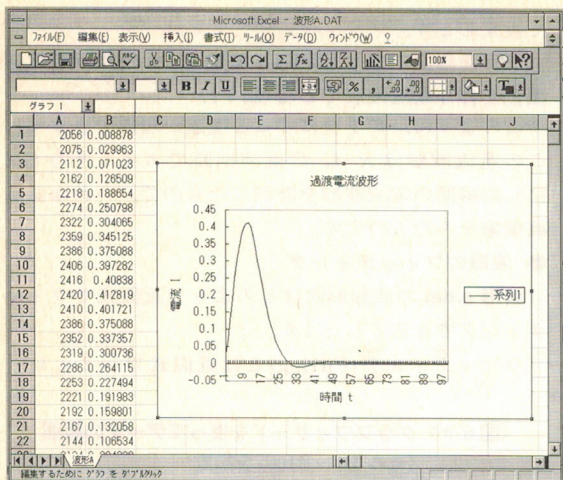
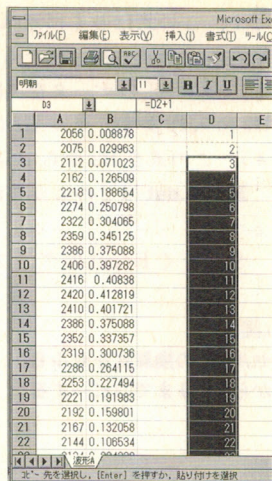
**波形 a.dat** の波形形状はどのような近似式でフィッティングできるでしょうか．

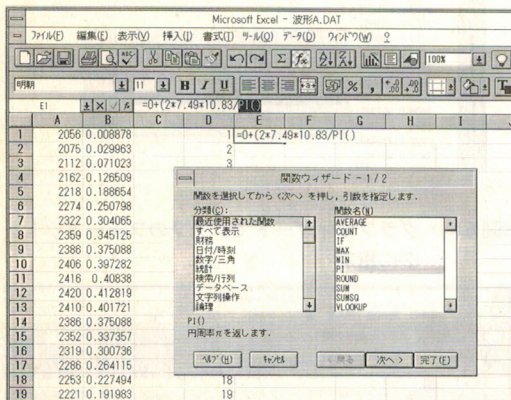
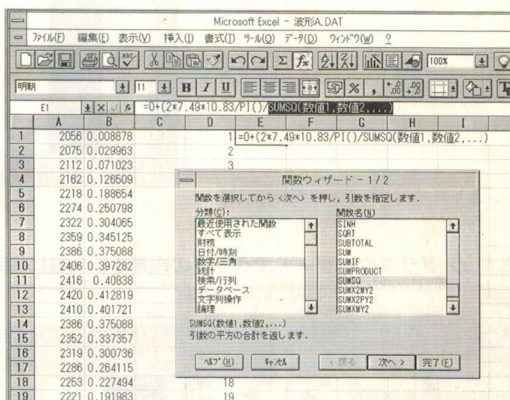
フィッティング(Fitting)とは近似式でシミュレー

〈図5〉
セルB1の換算式をセルB2からB640までコピーする

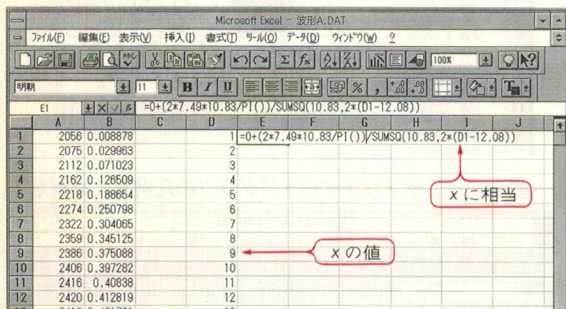
〈図6(a)〉 グラフウィザードを使ってデータの範囲(セルB1～B100)を指定

〈図6(b)〉 グラフウィザードでグラフの種類を折れ線に指定

〈図6(c)〉 グラフウィザードでグラフのフォーマットを(2)に指定

〈図6(d)〉 グラフウィザードでグラフのサンプルを確認

〈図6(e)〉 グラフウィザードでグラフの凡例などを指定

〈図7〉◀ 波形 a.dat の波形形状

〈図8〉▶ ローレンツの式の $x$ の値(1〜640)をセル D1〜D640 に入れる

〈図9(a)〉 セル E1 に関数ウィザードを使ってローレンツの式を入れる(PI を指定しているところ)

〈図9(b)〉 セル E1 に関数ウィザードを使ってローレンツの式を入れる(SUMSQ を指定しているところ)

〈図9(c)〉 セル E1 にローレンツの式をすべて入れ終わる

ションすることです．ただし，使う近似式が問題です．どのような近似式を使うかは経験と感にたよるところが大ですが，ここではフィッティング曲線としてローレンツの式を使うことにします．

ローレンツの式は $y$ を $x$ の関数として，次式で表せます．

$$y(x)=y_0+\frac{2w}{\pi}\frac{A}{w^2+4(x-x_0)^2}$$

$y_0$：初期値
$x_0$：ピーク中心
$w$：半値幅
$A$：ピーク下部の面積

**波形 a.dat** の場合のローレンツの式の各定数 $y_0$，$x_0$，$w$，$A$ の値は以下の値とします．

$y_0$＝**0**，$x_0$＝**12.08**，$w$＝**10.83**，$A$＝**7.49**

これらの値の根拠は，実は波形解析ソフト Origin を使って解析したときの結果[2]を利用していますが，慣れてくるとおのずから数値は見当がついてきます．

それではフィッティングをしてみましょう．

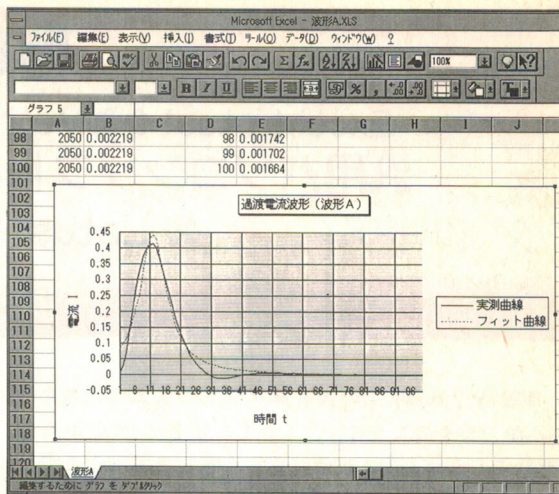
最初に，$x$ の値に **1, 2, 3, …, 640** を順次代入するために，**1** から **640** までの数値を Excel の D 列に入れておきます(**図8**)．ローレンツの式を代入します．セル E1 を使います(**図9**)．$\pi$ や $w^2+4*(x-x_0)^2$ などは関数ウィザードを使います．

=0+(2*7.49*10.83/PI())/SUMSQ

| | A | B | C | D | E |
|---|---|---|---|---|---|
| 1 | 2056 | 0.008878 | | 1 | 0.084886 |
| 2 | 2075 | 0.029963 | | 2 | 0.098604 |
| 3 | 2112 | 0.071023 | | 3 | 0.115508 |
| 4 | 2162 | 0.126509 | | 4 | 0.136458 |
| 5 | 2218 | 0.188654 | | 5 | 0.162497 |
| 6 | 2274 | 0.250798 | | 6 | 0.194756 |
| 7 | 2322 | 0.304065 | | 7 | 0.234182 |
| 8 | 2359 | 0.345125 | | 8 | 0.280846 |
| 9 | 2386 | 0.375088 | | 9 | 0.332661 |
| 10 | 2406 | 0.397282 | | 10 | 0.383675 |
| 11 | 2416 | 0.40838 | | 11 | 0.423441 |
| 12 | 2420 | 0.412819 | | 12 | 0.440189 |
| 13 | 2410 | 0.401721 | | 13 | 0.427932 |
| 14 | 2386 | 0.375088 | | 14 | 0.391114 |
| 15 | 2352 | 0.337357 | | 15 | 0.341099 |
| 16 | 2319 | 0.300736 | | 16 | 0.288891 |
| 17 | 2286 | 0.264115 | | 17 | 0.241182 |
| 18 | 2253 | 0.227494 | | 18 | 0.200565 |
| 19 | 2221 | 0.191983 | | 19 | 0.167211 |
| 20 | 2192 | 0.159801 | | 20 | 0.140253 |
| 21 | 2167 | 0.132058 | | 21 | 0.118568 |
| 22 | 2144 | 0.106534 | | 22 | 0.101075 |



〈図10〉◀
セルE1のローレンツの式をE2〜E640にコピーする

| | A | B | C | D | E |
|---|---|---|---|---|---|
| 98 | 2050 | 0.002219 | | 98 | 0.001742 |
| 99 | 2050 | 0.002219 | | 99 | 0.001702 |
| 100 | 2050 | 0.002219 | | 100 | 0.001664 |


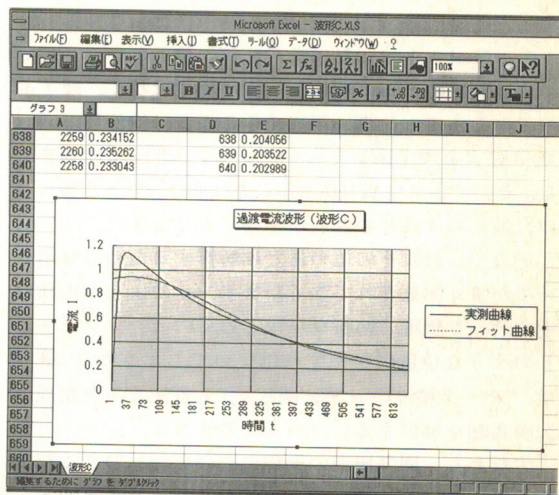

〈図12〉▶
実測曲線(波形 a.dat)とフィッティング曲線

=0+(2*7.49*10.83/PI())/SUMSQ

| | A | B | C | D | E |
|---|---|---|---|---|---|
| 1 | 2056 | 0.008878 | | 1 | 0.084886 |
| 2 | 2075 | 0.029963 | | 2 | 0.098604 |
| 3 | 2112 | 0.071023 | | 3 | 0.115508 |
| 4 | 2162 | 0.126509 | | 4 | 0.136458 |
| 5 | 2218 | 0.188654 | | 5 | 0.162497 |
| 6 | 2274 | 0.250798 | | 6 | 0.194756 |
| 7 | 2322 | 0.304065 | | 7 | 0.234182 |
| 8 | 2359 | 0.345125 | | 8 | 0.280846 |
| 9 | 2386 | 0.375088 | | 9 | 0.332661 |
| 10 | 2406 | 0.397282 | | 10 | 0.383675 |
| 11 | 2416 | 0.40838 | | 11 | 0.423441 |
| 12 | 2420 | 0.412819 | | 12 | 0.440189 |
| 13 | 2410 | 0.401721 | | 13 | 0.427932 |
| 14 | 2386 | 0.375088 | | 14 | 0.391114 |
| 15 | 2352 | 0.337357 | | 15 | 0.341099 |
| 16 | 2319 | 0.300736 | | 16 | 0.288891 |
| 17 | 2286 | 0.264115 | | 17 | 0.241182 |
| 18 | 2253 | 0.227494 | | 18 | 0.200565 |
| 19 | 2221 | 0.191983 | | 19 | 0.167211 |
| 20 | 2192 | 0.159801 | | 20 | 0.140253 |
| 21 | 2167 | 0.132058 | | 21 | 0.118563 |
| 22 | 2144 | 0.106534 | | 22 | 0.101075 |



〈図11〉◀
グラフ表示のためのデータの範囲(列Bと列E)を指定

| | A | B | C | D | E |
|---|---|---|---|---|---|
| 638 | 2259 | 0.234152 | | 638 | 0.204056 |
| 639 | 2260 | 0.235262 | | 639 | 0.203522 |
| 640 | 2258 | 0.233043 | | 640 | 0.202989 |



〈図13〉▶
実測曲線(波形 c.dat)とフィッティング曲線

すなわち，$\pi$ は **PI( )** を，$w^2+4*(x-x_0)^2$は **w²+(2*(x−x₀))²** として関数 **SUMSQ** を使います．式を入れ終わるとセルに計算結果が出てきます．この式をセル **E2** から **E640** のコピーします(**図10**)．

元のデータにうまくフィッティングできましたか？

グラフにしてみましょう．B列が元のデータ(ASCIIデータ)でE列は計算値です．これらを選択して強調表示します(**図11**)．グラフウィザードでグラフの書式指定します．結果は**図12**のようになりました．図中，実線が実測曲線で，点線がフィッティング曲線です．ほぼ“フィット”しました．

波形Cについてはどうでしょう．途中の計算などの説明は省略しますが，結果のグラフは**図13**です．実線が波形Cで，点線がローレンツの式を用いて計算したフィッティング曲線です．定数 $x_0$，$w$，$A$ の値は，

$y_0=0$，$x_0=41.01$，$w=628.91$，$A=928.15$

です．波形Cについてもおおむね“フィット”させることができました．

波形フィッティングは波形を解析するための一つの手段です．この事例では波形の中で大きなピークが一だけあるとして計算しましたが，波形が大きく振動してピークが複数ある場合には，次の多重ローレンツの式でフィッティングさせることが可能です．

多重ローレンツの式とは，

$$y(x)=y_0+\frac{2\,w_1}{\pi}\frac{A_1}{4(x-x_{01})^2}+\frac{2\,w_2}{\pi}\frac{A_2}{4(x-x_{02})^2}+\cdots+\frac{2\,w_n}{\pi}\frac{A_n}{4(x-x_{0n})^2}$$

です．先のローレンツの式は多重ローレンツの式の **n=1** の場合です．振動する波形のフィッティングを検討する場合には，多重ローレンツの式を使ってみるのも有効であると考えられます．

◆ **参考文献** ◆


(1) Peter A. Jansson；Deconvolution with Application in Spectroscopy，ACADEMIC PRESS，INC.

(2) 臼田；Originを使った波形シミュレーション，トラ技コンピュータ5月号，1995年，CQ出版㈱.

# 第6章

Excel, Lotus1-2-3, Mathematica を利用するノウハウ

# 汎用パソコンソフトを計測に活用する方法

## ② Lotus1-2-3を活用する

伊藤　敏

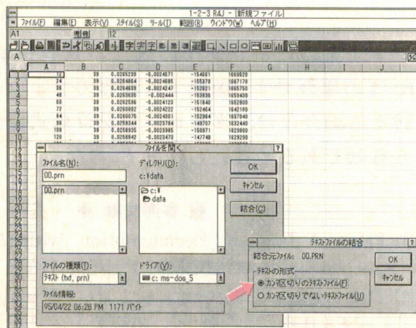
MS-Windows上の表計算ソフトの人気は「エクセル」が一番手のようです．しかし，筆者はDOSからの移行でLotus 1-2-3が使いやすく感じます．ここでは，Lotus 1-2-3 for Winを用いてデータの整理，さらには解析，シミュレーションを試みてみましょう．

表計算ソフトは，単に営業関係の整理とか，学校での成績の管理とかに利用されるだけではなく，理工系の人にとっても仕事に利用できることは拙著「Lotus 1-2-3による理工系シミュレーション入門」に述べました．そして，MS-Windows上ではDOSの上で持っていた厳しいメモリの制限が少なくなりました．

一方で，計測その他のあとに解析するための専用ツールが多く開発されてきています．そのような現状の中でデータ整理，解析ツールとしての表計算ソフトはどのような位置付けをすればよいでしょうか？ 筆者は，「データ整理の方法を探る」，「ある決まった整理方法の真髄を理解する」ツールと考えます．

すなわち，開発研究の過程においては，できるだけ柔軟にいろいろな整理方法や解析を試みるべきです．そのようなときには，ある程度操作になれ，かつ何をしているかが見える表計算ソフトが有効でしょう．また，何をしているかが見える表計算ソフト上での解析は，教育的にも重要な位置づけをもっています．

### 数値データをLotus 1-2-3に読み込む

ずいぶん，重い前書きになってしまいました．多くの整理・解析するデータはテキストファイルの形で提供されると思います．そこで，これらテキストファイル形式のデータを表計算ソフトに取り込むことから始めます．テキスト形式といっても，

**123.0, 234.1, 454.5, 98.0, 786.9, 238.8**

のように各データがカンマで区切られている場合と，

**123.0　234.1　454.5　98.0　786.9　238.8**

空白やタブで区切られている場合があります．いずれの場合もテキストファイルを数値データとして読み込むには，

- ▶**ファイル(F)**　▶**開く(O)**
- ▶ファイルの種類で**テキスト(txt, prn)**を選択
- ▶ファイルの選択をして　▶**結合(C)**
- ▶ここで**カンマ区切りのテキストファイル(F)**を選択する．

の手順を踏みます．注意するべき点は数値データがカンマでなく空白やタブで区切られている場合でも，数値データとして読み込むには「カンマ区切り」を選択することです．一方，データを文字データとして読み込む場合はカンマ区切りでないテキストファイル(U)を選びます(図1)．

Lotus 1-2-3 for Winには，DOS上のLotus 1-2-3の操作手順をそのまま使える方法が残されています．「/」キーを押すと**図2**のようにDOSと同じメニューが現れ，あとはそのままDOS上でのLotus 1-2-3と同じように操作できます．テキストファイルを数値データとして読み込むには，

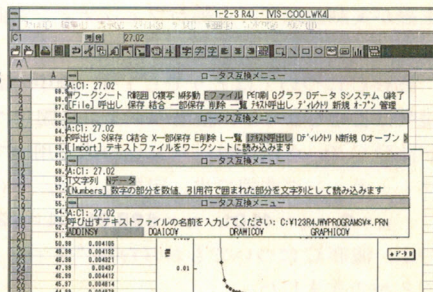
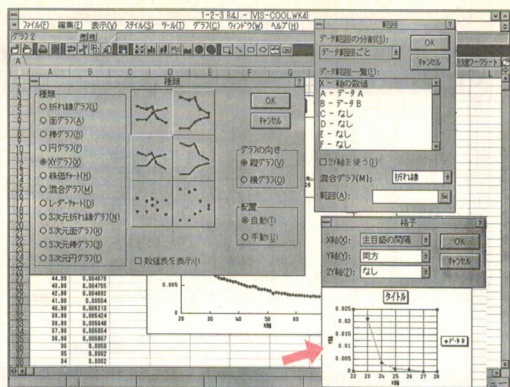
**/：メニュー　F：ファイル　I：テキスト呼出し**

〈図1〉データを数値データとして読み込む

〈図2〉DOS上のLotus 1-2-3との互換モードを使う

〈図3〉グラフの設定

〈図4〉物性量の時間変化

〈図5〉物性量の時間変化の解析

**N：データ**

の手順です．

## グラフ化と検討

ではつぎに，読み込んだデータをグラフにしましょう．いろいろ考えごとをするには，数値を眺めるよりは表にしたほうが，さらにグラフにしたほうが考えが進みます．数値データをグラフにするには，

▶**ツール(T)**

▶**グラフ(G)**

を選択して，グラフを表示する位置と大きさを決めます．その後，グラフの種類とデータ範囲の選択をして，グラフを表示します．その様子を**図3**に示しました．

時間変化をする物性量(粘度)について，LabVIEW(後述)の管理のもとGPIBを用いて測定した結果を表計算ソフトに取り込んで処理をする例を説明します．もっともデータはテキストファイルになっていますから，どのような実験系から取り込んでも同じ処理ですみます．

四つのデータをそれぞれ最初に述べた方法でLotus 1-2-3に取り込み，グラフを描きました．**図4**にそれを示します．ここで，A列が時間，C列が粘度のデータです．

このグラフから，粘度が指数関数的に変化をし，ある所からほぼ一定の値に収斂するように見えます．そこで，**図4**の時間変化の各グラフから，スタートの値と最小値の差を取り，各値の比をログでプロットしなおします．ここでは，C列とD列の間に新たに列を挿入して，C列のスタートの値と最小値との比をD列に計算しました．D列には，

**$LOG( ( C2 -@MIN($C$2..$C$23) )/( $C$2 -@MIN($C$2..$C$23) ) )**

式：**log( ( 値 － 最小値 )/ 変化幅 )**

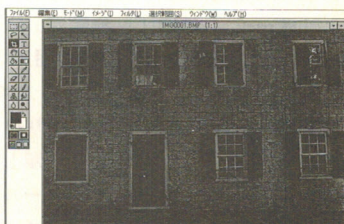
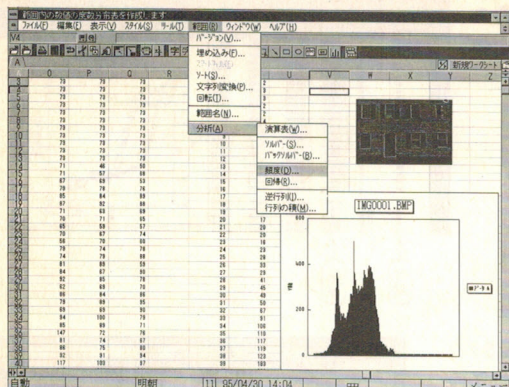
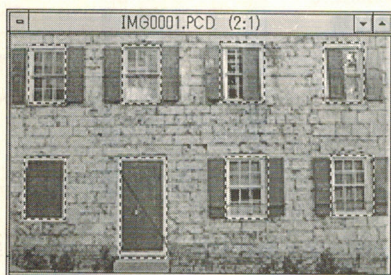
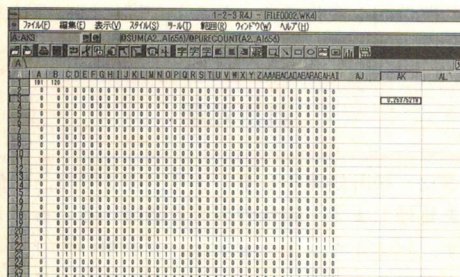
が入っています．その結果を**図5**に示します．

この**図5**より，この物性量の時間変化は，ある一定の時間まで変化が起こらず，一度変化し始めると条件が異なってもほぼ一定の傾きで変化が起きていることを示していることがわかります．また，当初の予想通り指数関数的な変化であったことが図にはっきりと示されています．

## 統計処理

つぎに，統計処理をしましょう．ここでは，画像データを処理します．画像の入力方法はいろいろ考えられます．ビデオキャプチャ・ボードを用いたり，スキャナを用いるのもよいでしょう．ここでは，最初に，PhotoCDからのデータとビデオキャプチャ・ボードからの画像を，PhotoShopでMS-Windowsのグレースケール画像のビットマップ・ファイルに変換します．そして，第8章のビデオキャプチャ・ボードの項で述べる**BMP2NUM.EXE**により数値化したファイルを表計算ソフトに取り込み，強度分布グラフを作ります．

強度分布は画像処理ソフトに同様な機能が内蔵され

〈図6〉PhotoCDサンプル画像IMG0001.PCDのPhotoShopによる処理

〈図7〉IMG0001グレイスケール画像の度数分布図

〈図9〉
範囲選択

〈図11〉面積比を求める

ていますが，自分自身で理解を深めるためにも，表計算ソフト上で処理をしました．

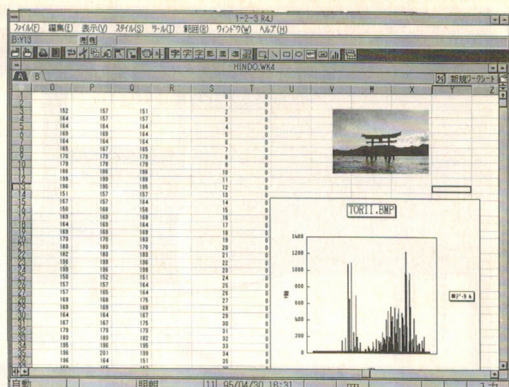
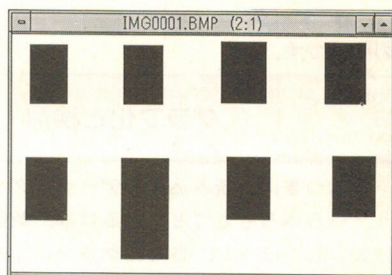
用いた画像は，PhotoCDに付属しているサンプル画像の**IMG0001.PCD**の192×128の画像を一部カットして用いました．その画像を**図6**に示します．

この画像をBMPのグレースケールに変化をして数値化をします．そのファイルをLotus 1-2-3に取り込んで，統計分析の一つである度数分布を計算します．

**▶範囲(R)　▶分析(A)　▶頻度(D)**

を選択して，度数分布を取る「対象範囲」と「階級範囲」を設定します．ここでは「対象範囲」として，画像データそのものを取ります．「階級範囲」はグレースケールの数値範囲である，0から255までの値をLotus 1-2-3の埋め込み機能を用いて作り指定します．

その結果をグラフ化したものを**図7**に示します．縦軸が出現度数を示し，横軸がグレースケールの明るさを示します．横軸が大きい程明るくなります．

〈図8〉鳥居のグレイスケール画像の度数分布図

〈図10〉
図9の絵を
2値化

同様に，鳥居を撮影した画像を処理しました．その結果を**図8**に示します．まったく異なる画像ですから，**図7**と比べると度数分布図が大きく異なります．元の画像をLotus 1-2-3のシート上に表示してあります．

## その他の処理

画像を処理して2値化し，面積を表計算ソフト上で求めましょう．

ビデオキャプチャ・ボードの章でも紹介をしますが，ここでは別の例で具体的な手順を見てみましょう．**図6**で示したPhotoCDのサンプル画像の窓の部分の面積比を求めます．**図9**のようにPhotoShop上で窓の部分を範囲選択します．

選択された範囲を黒にその他の範囲を白に2値化します．**図10**のように味気ない図になりました．

この図を数値化して，Lotus 1-2-3に取り込みました．この場合は白色が0，黒が1になるようにしました．したがって，画像データのはいった範囲のすべての和を範囲の全セル数で割ると黒色の部分の面積比が求まります(**図11**)．セルAK3に，

**@SUM(A2..AI656)/@PURECOUNT(A2..AI656)**

**@SUM：指定範囲の全合計**

**@PURECOUNT：指定範囲の数値を含むセルの数**

を代入し面積比0.25375が与えられています．

今回は単純な形をした例で示しましたが，形が複雑になればなるほど，この方法は有効になるでしょう．

# 第6章

## Excel, Lotus1-2-3, Mathematicaを利用するノウハウ
## 汎用パソコンソフトを計測に活用する方法
### ③ Mathematicaを活用する

井上祥史

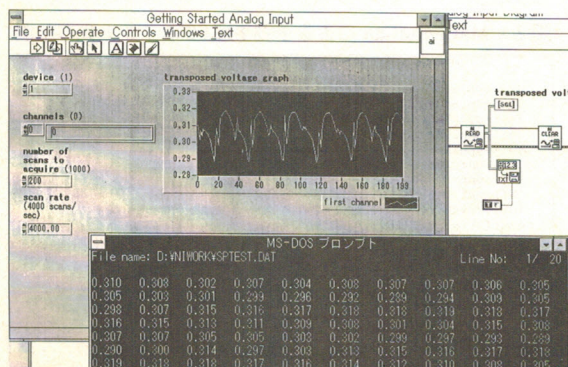
LabVIEWでは，ほかのアプリケーションとの動的リンクを取ることにより，ほかのアプリケーションをSubVIのように扱い，その機能をLabVIEWの中から利用することができます．

たとえば，Mathematicaとリンクするには，MathLinkという通信プログラムを用います．リンクが確立されると，LabVIEWからMathematicaを呼び出したり，逆にMathematicaからLabVIEWを呼び出すことができ，互いのアプリケーションの特徴を最大限に活用する環境を作ることができます．

Mathematicaの最大の特徴は，数式処理と美しいグラフ出力で，LabVIEWの特徴は計測データ収録と制御信号が出力可能なことです．LabVIEWには多くの強力な数値処理解析ツールが付属しており，その機能はMathematicaと同等かそれ以上です．

したがって，MathematicaとLabVIEWの併用を考える場合，LabVIEWでデータを収録し，Mathematicaでグラフの表示を行ったり数式による解析的な処理を行った後，得られた値をLabVIEWに返して制御出力信号を出すという形が考えられます．

MathLinkが利用できない場合は，これらの操作をファイルに書き出したり読み出したりしてつないでいきます．ここでは，ファイルの読み書きを通してLabVIEWからMathematicaを利用する例を見てみます．

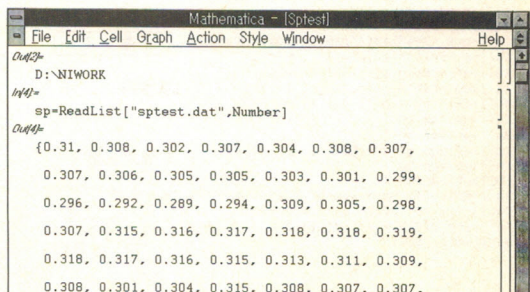
〈図1〉LabVIEWからデータをファイルに書き出す

## Mathematicaによるフーリエ変換

まず，最初はLabVIEWとMathematicaのスペクトル解析の結果を比較してみます．蛍光灯の光の強弱の信号をLabVIEWで波形表示すると同時に，ファイルを介してMathematicaに送り，Mathematicaでフーリエ変換してみます．外部からの信号をLabVIEWにそのまま取り込むには，データ収録ボード（アナログ-ディジタルI/Oボード）を用います．

使用するVIは，最も簡単に利用できるサンプルVI（Getting Started AnalogInput.VI）を用いました．取り込んだ配列データをファイルに書き出すのは，中間（intermediate）VIであるWrite To SpreadSheet File.VIをダイヤグラムに追加するだけで簡単にできます．このWrite To SpreadSheet File.VIは，時系列などの配列データを文字列に変換し，ワークシートのイメージのままファイルに書き出す働きをします．

図1の左のダイヤグラムに，I/Oボードから読み込んだデータをグラフ表示器とWrite To SpreadSheet File.VIとに配線している様子を示しています．MS-DOSプロンプト画面には，このようにして書き出したファイル名の中身のデータを示しています．

いったんファイルに書き出した後は，MathematicaやMathcadそして表計算などのアプリケーションで処理をするのは簡単です．図2は以下のような命令でMathematicaへ読み込んだ結果を，1次元要素のリス

〈図2〉ファイルからMathematicaにデータを読み込む

〈図 3〉 Mathematica による蛍光灯の光のフーリエ変換

〈図 4〉 自己回帰直線モデルによる予測式

次のような自己回帰直線モデルを考える.

$$y_{t+1}=b\,y_t+a\,y_{t-1}$$

このとき,

$$e=\Sigma(y_{t+1}-b\,y_t-a\,y_{t-1})^2$$

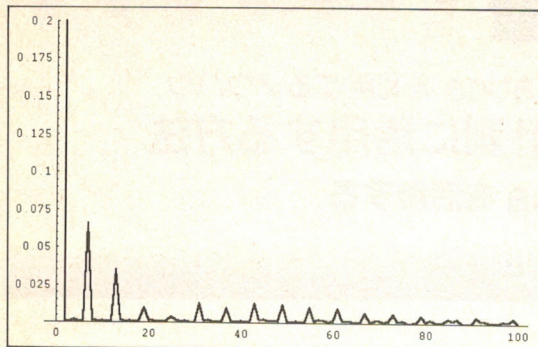
を最小にする条件,

$$\frac{\partial e}{\partial b}=0,\quad \frac{\partial e}{\partial a}=0$$

から,

$$\begin{cases} b\Sigma y_t^2+a\Sigma y_{t-1}\,y_t=\Sigma y_t\,y_{t+1} \\ b\Sigma y_{t-1}\,y_t+a\Sigma y_{t-1}^2=\Sigma y_{t-1}\,y_{t+1} \end{cases}$$

の $(a, b)$ に関する連立方程式を得る.

**図 5**, **図 6** では, $y_{t-1}$, $y_t$, $y_{t+1}$ をそれぞれ $p$, $n$, $f$(過去, 現在, 未来)の略号に対応させている.

トの並びとして示したものです.

**In[ ]=SetDirectory["d:¥¥niwork"]**

**Out[ ]=D:¥NIWORK**

**In[ ]=sp:=ReadList["sptest.dat", Number]**

もちろん, **図 1** のデータ列とまったく同一のリストになっています.

この **sp** というリストデータをフーリエ変換して絶対値の大きさを表示するには, 続いて次のような命令を Mathematica に打ち込みます.

**In[ ]=<<Calculus'FourierTransform'**

**In[ ]=fft:=Fourier[sp]**

**In[ ]=upfft:=Take[fft, 100]**

**In[ ]=ListPlot[Abs[upfft], PlotJoined->**
**True, PlotRange->{0.0,0.4}]**

**Take[ ]**は 200 個の **fft** の要素の中から最初の 100 個を取り出す命令で, スペクトルを拡大表示して見やすくするためのものです.

結果は**図 3** のようになって, 蛍光灯の光は多くの高周波成分を含んでいることがわかります.

**図 1** のフルスケールの 200 点は 200/4000=1/20 秒であることから, フーリエ変換の横軸の 1 点に相当する分解能は 20 Hz となります. したがって, 最初のピーク(6 点目)の周波数は 120 Hz になります. 以下 240 Hz(12 点目)360 Hz(18 点目)と続いています. この FFT のグラフを LabVIEW を使ったパワースペクトル(後出)と比較してみてください.

## Mathematica で予測する

LabVIEW の数値演算機能以外の計算が必要な場合には, C 言語などを用いてモジュールを作るかフォーミュラノードを利用するか, あるいは Mathematica などほかのアプリケーションを援用することになります. LabVIEW の解析ツールにない計算例として, 自己回帰モデルによる予測計算を Mathematica で行ってみます.

地震などの自然現象を, 観測しながら予測することはきわめて重要な課題で, 予測ができれば, それに基づいて適切な対応をとることができます. 制御系では予測に基づいた制御信号を出すことに相当し, 系を安定にそして安全に運転することができます.

たとえば, 風力発電をしてバッテリに電力を蓄え, 照明などの負荷を接続した系を考えてみます. 風速の測定をしていて, 今は風が吹いていないけれども今夜の照明のスイッチを予測に基づいて入れることができれば, より実用的な知能システムになることは明らかです.

LabVIEW を通して計測データを収録し, 解析・予

〈図 5〉 Mathematica による予測手順(1 ポイント先の予測)

```
In[]=SetDirectory["d:\\niwork"]
Out[]=D:\NIWORK
In[]=pre:=ReadList["predict.dat",Number]
In[]=p:=Partition[pre,2]
In[]=qn:=Table[p[[i,2]],{i,2,19}]
In[]=qp:=Table[p[[i,2]],{i,1,18}]
In[]=qf:=Table[p[[i,2]],{i,3,20}]
In[]=nn:=qn.qn
In[]=pn:=qp.qn
In[]=nf:=qn.qf
In[]=pp:=qp.qp
In[]=pf:=qp.qf
In[]=Solve[{b nn+a pn==nf,b pn+a pp==pf},{a,b}]
Out[]={{a->0.586818,b->0.429292}}
In[]=f=b p[[20,2]]+a p[[19,2]] /. %
Out[]={12.7687}
```


〈図 6〉 Mathematica による予測手順(2 ポイント先の予測)

```
In[]=dn:=Table[p[[i,2]],{i,4,18,2}]
In[]=dp:=Table[p[[i,2]],{i,2,16,2}]
In[]=df:=Table[p[[i,2]],{i,6,20,2}]
In[]=dnn:=dn.dn
In[]=dpn:=dp.dn
In[]=dnf:=dn.df
In[]=dpp:=dp.dp
In[]=dpf:=dp.df
In[]=Solve[{b dnn+a dpn==dnf,b dpn+a dpp==dpf},{a,b}]
Out[]={{a->0.379645,b->0.665744}}
In[]=f=b p[[20,2]]+a p[[18,2]] /. %
Out[]={11.6236}
In[]=qt:=Table[p[[i,2]],{i,1,20}]
In[]=qt2:=Join[qt,{12.7687,11.6236}]
In[]=ListPlot[qt2,PlotJoined->True]
```

1 点おきの配列
内積による和
t−2, t による
t+2 の予測(22 点目)
方程式の解
元の配列
予測値の結合
グラフ化

測をしてLabVIEWから制御出力することでこのような知能的な予測制御系ができあがることになります.

### ● 予測の手法

予測には様々な手法が知られています. 単純な外挿法から始まって相関・回帰分析による予測, パターン予測, ニューラルネットによる学習, そしてカオス性があればアトラクタの軌跡やフラクタル性からの予測などが考えられます. このうち, 最も基本的な自己回帰モデルによる予測を考えてみます.

簡単のため, LabVIEWからの仮想計測データが**predict.dat**というファイルにWrite To SpreadSheet File.VIを通して書き込まれているとします. Mathematicaでその仮想データを読み込んで, **図4**の直線の自己回帰式にしたがって命令を書いていきます.

この直線の自己回帰モデルは, 時系列データの連続した3点の間の誤差を最小にする関係式を求めるものです. 仮想データは20個あるものとし, **0, data0, 1, data1, 2, data2, …, 19, data19**のような配列で書き込んであるものとしました.

この仮想データを使って, **図5**に1点先(**t+1**；21点目)の予測値を求める手順を, **図6**に2点先(**t+2**；22点目)の予測値を求める手順を示しました.

**図7**に, 現在までのデータ(1～20点)とこのようにして得られた予測値(21および22点)とを, 同時にグラフに示しました. 実際には, これらの予測値をLabVIEWに返して, 制御信号を出して行くことになります.

〈図7〉現在までのデータ(1～20点)と予測値(21, 21点)

**図7**の結果は, 明らかに最小2乗法の外挿などとは異なり, 折れ曲がった確からしい予測値を示しています.

精度を犠牲にすれば, このような方法で何点先の予測値でも求めることができます. 予測値は, もちろん実測によってその確からしさを検証するもので, いかに精度のよい予測を行うことができるか, 実測に照らし合わせながら盛んに研究されています.

# 第7章

## 多機能な計測専用ツールLabVIEWの使い方
# LabVIEWによるシミュレーションの方法

臼田昭司

## LabVIEWとはどんなツール

計測・制御用ツールには，用途に応じたいろいろな種類のものがあります．例えば，MATLAB(1)，SIMULINK(2)，MATRIX(3)…などです．LabVIEW(4)もこの分野ではよく知られたツールです．

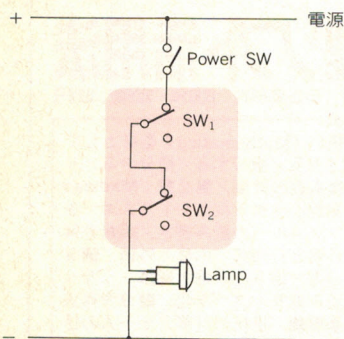
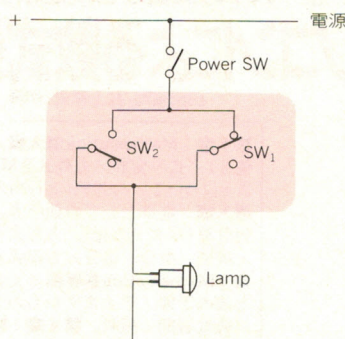
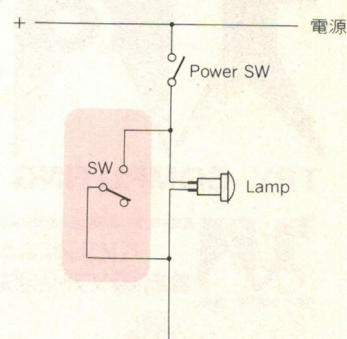
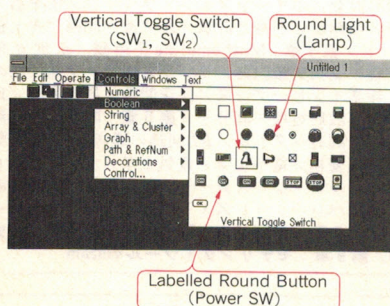
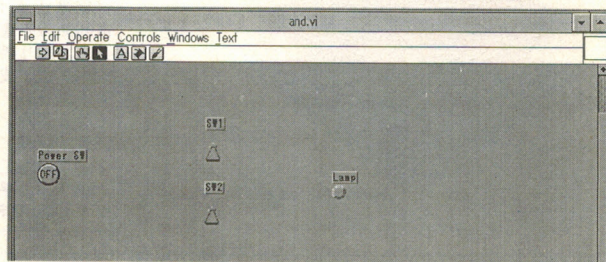
このツールは，米国の計測・制御メーカーであるNational Instruments社(4)の製品です．ハード面のノウハウをバックグラウンドに，自社の計測制御ボードを支援するソフトとして開発されたようです．

その後，バージョンアップを経て，現在はソフトウェアの単独商品として，国内では同社の日本法人から販売・サポートされています．ハード面を支援するソフトといっても，実は計測・制御にかかわる多くの機能がLabVIEWの中に盛り込まれています．ここで紹介するシミュレーションなども，これらの機能を使って比較的簡単に行うことができます．

### ● LabVIEWの概要

最初に，"LabVIEWとはどんなツール"という質問に対する簡単な答えとして，制御ではよく知られた論理回路の例を紹介します．

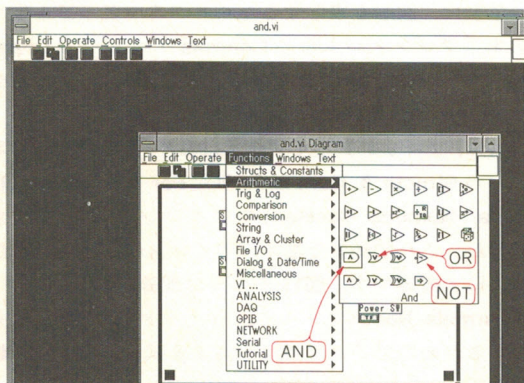
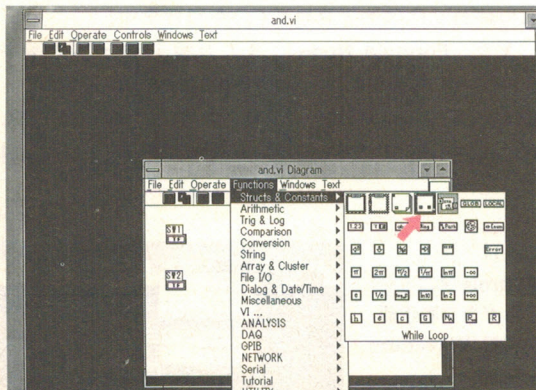
**図1**は，それぞれブール代数でおなじみのAND(論理積)，OR(論理和)，NOT(否定)の回路例です．この回路を構成している部品[LabVIEWでは**Boolean**(ブール)制御器という]は，電源をON/OFFするパワースイッチ(**Power SW**)と，論理動作を確認するトグルスイッチ(**SW1, SW2**)とランプ(**Lamp**)です．

これらの部品の作成については，LabVIEWではそれぞれアイコン**Labelled Round Button**，**Vertical Toggle Switch**，**Round Light**をフロントパネルの**Controls**メニューから選択します(**図2**)．

〈図1(a)〉 AND回路　〈図1(b)〉 OR回路　〈図1(c)〉 NOT回路

〈図2〉 Controlsメニューから Boolean制御器を選択

〈図3〉 フロントパネルでBoolean制御器を配置

まず最初に，AND回路を作ってみましょう．

LabVIEWでは『フロントパネル』と『ブロックダイヤグラム』を用いてシステムを構築します．詳細についてはマニュアルを参照してください．

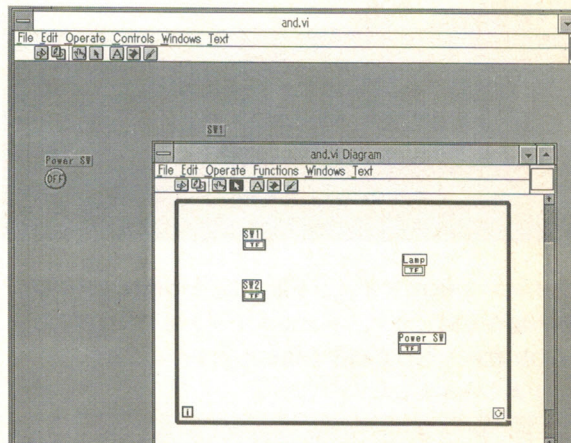
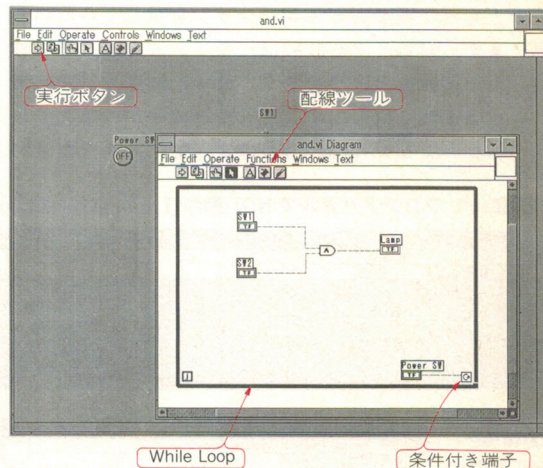
フロントパネルで上記の部品類を配置します(図3)．次に，Windowsメニューから**Show Diagram**を選択しブロックダイヤグラムを開きます(図4)．図には各部品の端子(接続端子**TF**)が表示されています．ここで，**Functions**メニューの**Arithmetic**と**Structs & Constants**からそれぞれ**AND**アイコンと**While Loop**アイコンを選択します(図5)．

次に，**While Loop**アイコンをマウス操作で拡大し，この中で各端子と**AND**アイコンを図6のように配線ツールで接続します．**While loop**アイコンは繰り返し動作を行わせるプログラムアイコンで，条件付き端子をもっています．図6ではパワースイッチ(**Power SW**)がこの端子に接続されています．パワースイッチを押してはじめて，トグルスイッチの繰り返しON/OFF動作が可能になるという意味です．

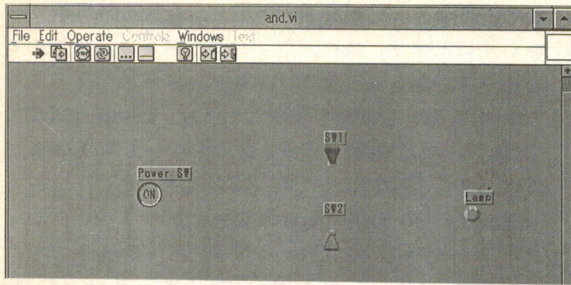
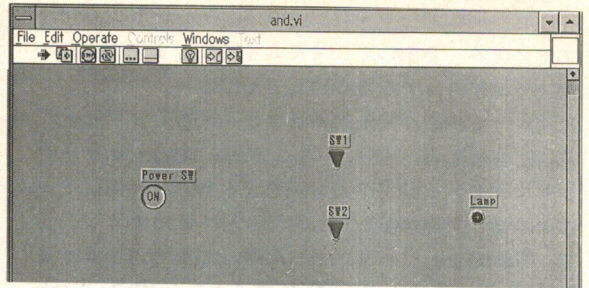
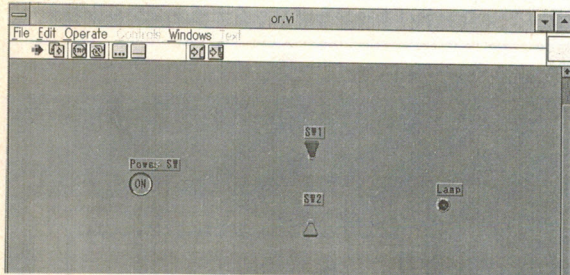
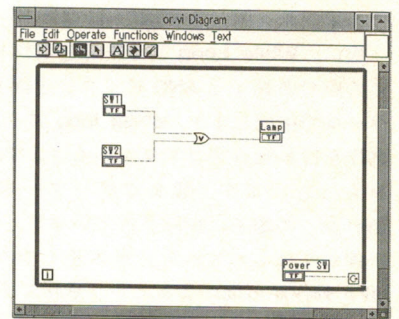
〈図5(a)〉ブロックダイヤグラムのFunctionsメニューからANDアイコンを選択

〈図5(b)〉ブロックダイヤグラムのFunctionsメニューからWhile Loopアイコンを選択

それでは動作をさせてみましょう．**表1**はANDの真理値表です．フロントパネルで，最初に**Power SW**をONにします．次に，ウィンドウメニューの下のモードボタンから実行ボタンをクリックします．

**SW1**を**ON**，**SW2**を**OFF**にします．**Lamp**は消灯です(図7)．次に**SW1**と**SW2**を**ON**にします．**Lamp**は点灯しました(図8)．こんな具合です．

図9は，OR回路の場合のフロントパネルとブロックダイヤグラムで，図10はNOT回路の場合です．**Functions**メニューの**Arithmetic**から，それぞれ**OR**

〈図4〉フロントパネルからブロックダイヤグラムを開く

〈図6〉編集後のブロックダイヤグラム(AND回路)

〈表1〉ANDの真理値表

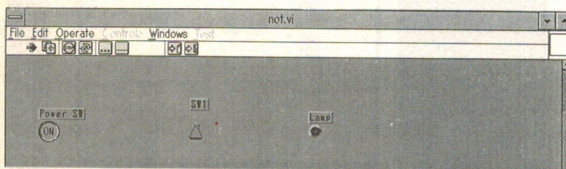
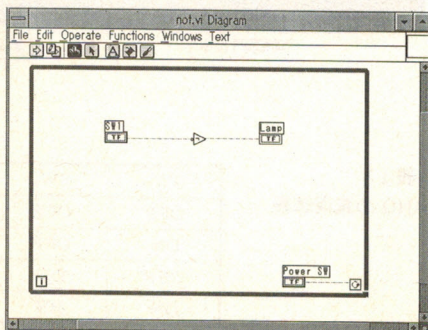
| $SW_1$ | $SW_2$ | Lamp |
|---|---|---|
| OFF | OFF | 消灯 |
| ON | OFF | 消灯 |
| OFF | ON | 消灯 |
| ON | ON | 点灯 |

〈図7〉フロントパネルでAND動作(Lamp消灯の場合)

〈図8〉フロントパネルでAND動作(Lamp点灯の場合)

〈図9(a)〉フロントパネルでOR動作(Lamp点灯の場合)

〈図9(b)〉
OR回路のブロックダイヤグラム

アイコンと**NOT**アイコンを選択し，その他はAND回路の場合と同じです．トグルスイッチをON/OFFしてOR動作とNOT動作を確かめます．

うまくいきました！

## 計算で波形シミュレーション

ここでは，波形シミュレーションを試してみたいと思います．対象とする波形は，サイン波形，ノイズ波形，カオス波形の3種類です．サイン波形やノイズ波形などは実験で作り出すこともできますが，ここではシミュレーション(模擬実験)によってこのような波形を作り出してみようというわけです．

〈図10(a)〉フロントパネルでNOT動作(Lamp点灯の場合)

〈図10(b)〉
NOT回路のブロックダイヤグラム

**(1) サイン波形**

最初に，波形を表示するためのグラフ表示器を作ります．フロントパネルの**Controls**メニューの**Graph**から**Waveform Graph**を選択します(**図11**)．このグラフ表示器に**Sin Wave**と名前を付けます．

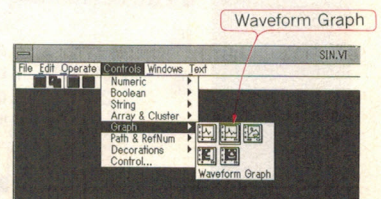
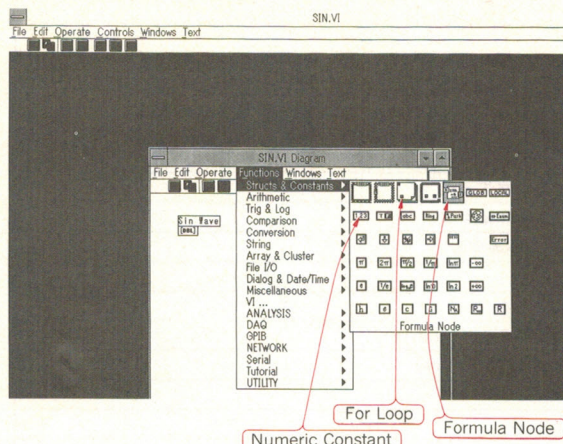
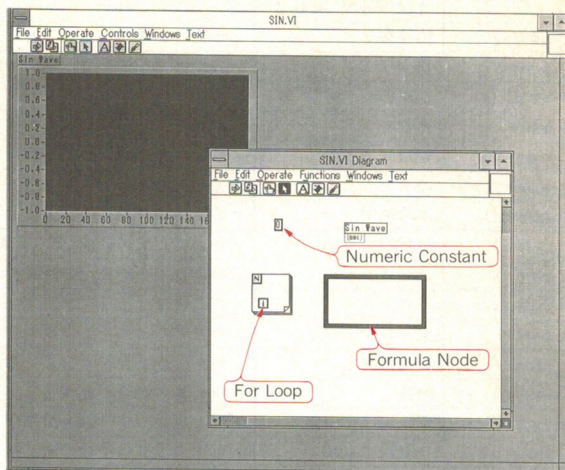
次に，ブロックダイヤグラムに移り，**Functions**メニューの**Structs & Constants**から**For Loop**，**Formula Node**，**Numeric Constant**のアイコンを選択します(**図12**)．これで必要な端子，アイコンがそろいました．いよいよ配線です．**図13**のように配線(編集)します．

**Formula Node**(フォーミュラノード)は，数式を実行するためのアイコンで，入力する式は**y**を**i**の関数として，

$$y(i)=\sin(0.5*i)$$

で与えています．これを**Formula Node**の中にテキストベースのプログラミング言語として書き込みます．式の終わりにはセミコロン(;)をつけます．

また，**Formula Node**には入力端子(**i**)と出力端子(**y**)を付け加えておきます．**For Loop**アイコンで，**Numeric Constant**を**200**に設定し，**i**を**0**から**200−1=199**まで繰り返し計算し(200回計算し)，得

〈図11〉
Controlsメニューからグラフ表示器を選択(Waveform Graph)

〈図12(a)〉 Functionsメニューから Formula Nodeなどのアイコンを選択

〈図12(b)〉 Formula Nodeなどのアイコンを選択後のブロックダイヤグラム

られたyの値を**Sin Wave**の端子(**DBL**)に入力して波形を表示させます．

さっそく，フロントパネルにもどって実行してみましょう．実行ボタンをクリックしてみると，**図14**のようになりました．

### (2) ノイズ波形

フロントパネルとブロックダイヤグラムの設計は，サイン波形の場合とまったく同じです．**Formula Node**に入力する数式のみが異なります．数式は**y**を**i**の関数として，

$$y=0.5*(\sin(0.5*i)+2*(\mathrm{rand}()-0.5))$$

です．この式は，サイン波形**sin (0.5＊i)**にノイズとして乱数**rand( )**を与えたものです．**y**が**＋1**から**－1**の間の値を取るように適当な係数を掛けたり，引いたりしています．なお，**rand( )**はLabVIEWのフォーミュラノード関数(**表2**)の一つです．

〈図13〉 編集後のブロックダイヤグラム(サイン波形の場合)

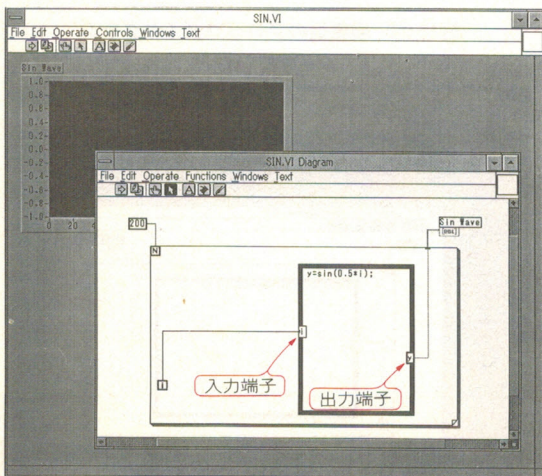



**図15**は，フロントパネル(実行画面)とブロックダイヤグラムです．**図15**(a)の波形は，いかにもノイズ波形のように見えます．

### (3) カオス波形

フロントパネルとブロックダイヤグラムについては，前の2例とほとんど同じですが，**Formula Node**に与える式はこれまでと少し異なり，初期値を与えた時系列の式となります．また，アイコン**For Loop**にはシフトレジスタ(**Add Shift Register**)を追加します(**図16**)．

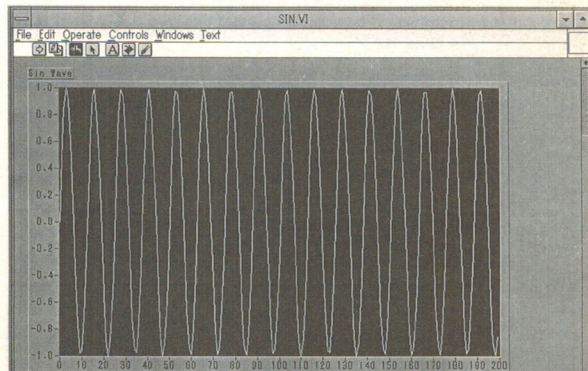
**図17**(a)(実行画面)と**図17**(b)は，それぞれフロントパネルとブロックダイヤグラムです．ブロックダイヤグラムの**Formula Node**には，計算式としてロジスティック写像で知られた，

$$y=4.0*i*(1-i)$$

を与えています．初期値は**Numeric Constant**アイコンで**0.1**を与えています．この式で，シフトレジスタは次のように働きます．

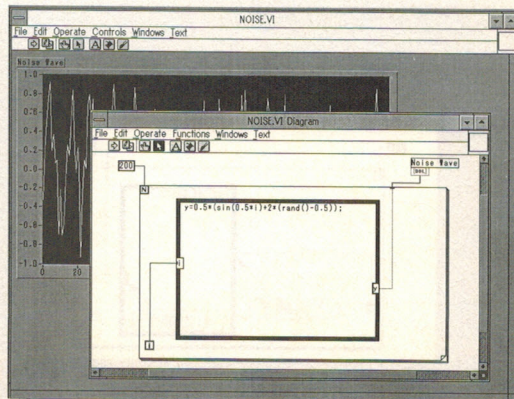
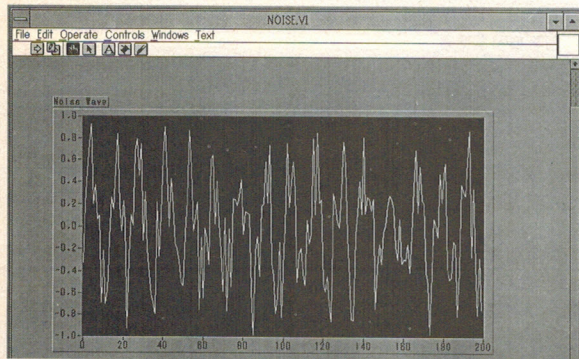
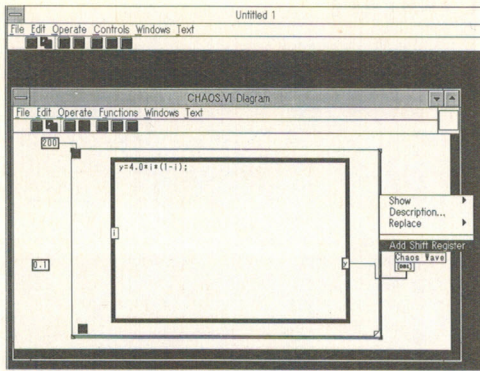
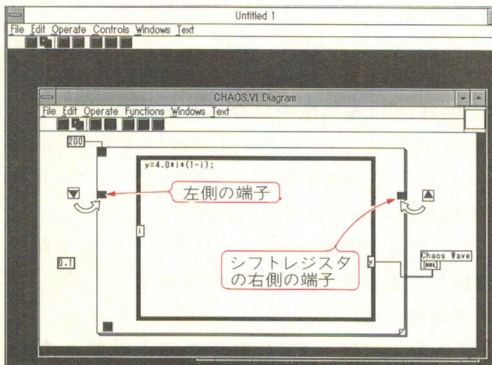
初期値を $y_0\ (=0.1)$ とすると，

$$y_1=4.0*y_0*(1-y_0)$$

〈図14〉 フロントパネルでサイン波形をシミュレーション

〈表2〉LabVIEW のフォーミュラノード関数一覧

| 関数 | 対応する LabVIEW 関数名 | 説明 |
|---|---|---|
| abs(x) | 絶対値 | x の絶対値を返す |
| acos(x) | 逆コサイン | x の逆コサインをラジアンで計算する |
| acosh(x) | 逆双曲コサイン | x の逆双曲コサインをラジアンで計算する |
| asin(x) | 逆サイン | x の逆サインをラジアンで計算する |
| asinh(x) | 逆双曲サイン | x の逆双曲サインをラジアンで計算する |
| atan(x, y) | 逆タンジェント | y/x の逆タンジェントをラジアンで計算する |
| atanh(x) | 逆双曲タンジェント | x の逆双曲タンジェントをラジアンで計算する |
| ceil(x) | ＋無限に丸め | x を次に高位の整数(最小整数≧x)に調整する |
| cos(x) | コサイン | x のコサインをラジアンで計算する |
| cosh(x) | 双曲コサイン | x の双曲コサインをラジアンで計算する |
| cot(x) | コタンジェント | x のコタンジェントをラジアンで計算する〔1/tan(x)〕 |
| csc(x) | コセカント | x のコセカントをラジアンで計算する〔1/sin(x)〕 |
| exp(x) | 指数 | e の x 乗の値を計算する |
| expm1(x) | 指数(Arg)－1 | e の x 乗引く1の値を計算する($e^x-1$) |
| floor(x) | －無限に丸め | x を次に下位の整数に切り捨てる(最大整数≦x) |
| getexp(x) | 仮数と指数 | 指数を返す |
| getman(x) | 仮数と指数 | 仮数を返す |
| intrz(x) | 0に調整 | x を x とゼロの間の最も近い整数に調整する |
| ln(x) | 自然対数 | x の自然対数(e を底とする)を計算する |
| lnpl(x) | 自然対数(Arg＋1) | (x＋1)の自然対数を計算する |
| log(x) | 10を底とする対数 | x の対数(10を底とする)を計算する |
| log2(x) | 2を底とする対数 | x の対数(2を底とする)を計算する |
| max(x, y) | 最大と最小 | x と y を比較し，大きいほうの値を返す |
| min(x, y) | 最大と最小 | x と y み比較し，小さいほうの値を返す |
| mod(x, y) | 商と余り | 商が－無限に丸められた場合の，x/y の余りを計算する |
| rand() | 乱数(0～1) | 0から1まで(0と1は含まない)の浮動小数点数を生成する |
| rem(x, y) | 余り | 商が最も近い整数に調整されることを除き，mod と同じ |
| sec(x) | セカント | x ラジアンのセカントを計算する〔1/cos(x)〕 |
| sign(x) | 符号 | x が0よりも大きい場合は1を返し，x が0に等しい場合は0を返し，x が0未満の場合は－1を返す |
| sin(x) | サイン | x ラジアンのサインを計算する |
| sinc(x) | Sinc | x のサイン割る x ラジアンを計算する〔sin(x)/x〕 |
| sinh(x) | 双曲サイン | x の双曲サインをラジアンで計算する |
| sqrt(x) | 平方根 | x の平方根を計算する |
| tan(x) | タンジェント | x のタンジェントをラジアンで計算する |
| tanh(x) | 双曲タンジェント | x の双曲タンジェントをラジアンで計算する |
| x^y | $x^y$ | x を y 乗した値を計算する |

〈図15(a)〉フロントパネルでノイズ波形をシミュレーション

〈図15(b)〉編集後のブロックダイヤグラム
(ノイズ波形の場合)

〈図16(a)〉 カオス波形の場合のブロックダイヤグラムを編集(Add Shift Register を選択)

〈図16(b)〉 カオス波形の場合のブロックダイアグラムを編集(シフトレジスタの端子が追加される)

となります．次に，この $y_1$を使って同様の計算します．

$$y_2=4.0*y_1*(1-y_1)$$

同じようにして，次は $y_2$を使って計算します．

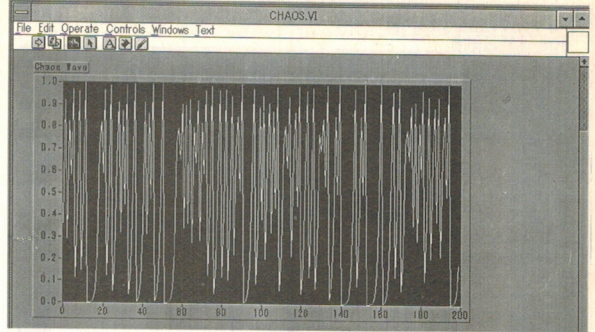
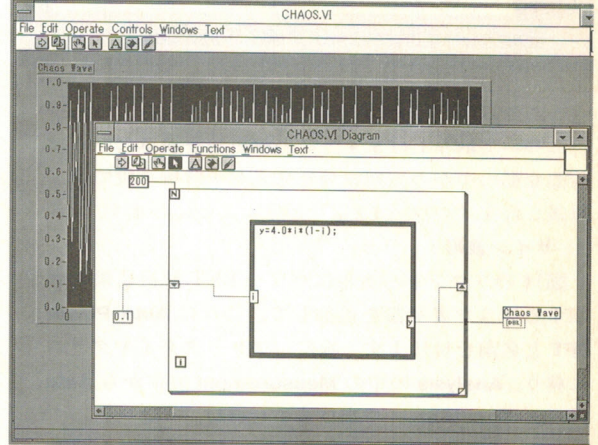
$$y_3=4.0*y_2*(1-y_2)$$

以下，**For Loop** アイコンで指定した回数(200回)の計算を繰り返します．

すなわち，シフトレジスタの右側の端子(上向きの矢印の矩形)に最初の計算結果($y_1$)が記憶され，これを左側の端子にシフトします．次の計算開始のときに，このデータ($y_1$)を使って $y_2$の計算をします．これを繰り返します．電子回路でいうシフトレジスタと同じ原理です．同じ名前が付けられたゆえんです．

実行結果の図17(a)の波形は，カオス波形(カオス時系列)と呼ばれているものです．サイン波形とも違うし，ノイズ波形とも違うし，なにか“混沌”としています．

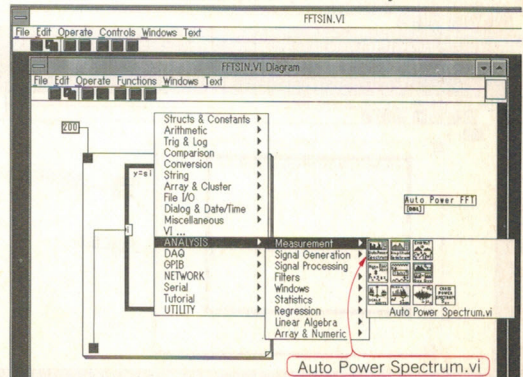
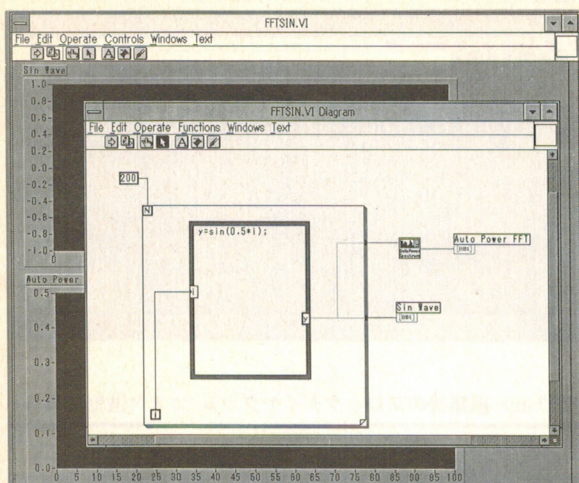
なお，カオスについては関係の書籍[5]がたくさん出ていますので，機会がありましたなら一度ひもといてみてください．

〈図17(a)〉 フロントパネルでカオス波形をシミュレーション

〈図17(b)〉 編集後のブロックダイヤグラム(カオス波形の場合)

## シミュレーション波形を解析する

前項の方法で得られた3種類の波形(サイン波形，ノイズ波形，カオス波形)について，波形解析をしてみます．LabVIEW には波形解析ツールが，**Functions** メニューの中の **Analysis** にたくさん含まれています．

〈図18〉 ブロックダイヤグラムでパワースペクトルアイコンを選択(Auto Power Spectrum.vi)

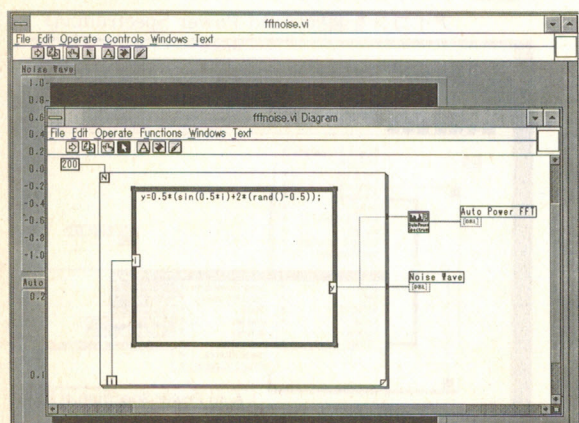
〈図19(a)〉 編集後のブロックダイヤグラム(サイン波形の場合)

〈図19(b)〉 サイン波形とそのパワーFFT

ここでは，パワースペクトル(Power Spectrum)解析について取り上げます．この解析法を簡単にいえば，時間変動の中から周期成分を見いだす解析手段です．以後これを“パワーFFT”と呼ぶことにします．

**(1) サイン波形**

**図14**のフロントパネルにパワーFFT解析結果を表示するグラフ表示器を追加して，これに**Auto Power FFT**と名前を付けます．次に，ブロックダイヤグラムに移り，**Analysis**の中の**Measurement**の中から**Auto Power Spectrum.vi**を選びます(**図18**)．

**Auto Power Spectrum**と書かれたアイコンが現れました．あとはグラフ表示器(**Auto Power FFT**)の端子や**Formula Node**の出力端子(**y**)などを配線して編集完了です〔**図19(a)**〕．

実行してみましょう．フロントパネルにもどって実行ボタンをクリックすると，**図19(b)**が得られました．パワーFFTには鋭いピークが得られますが，もともとサイン波形は周期関数ですから当然の結果です．

**(2) ノイズ波形**

サイン波形の場合と同様，**図15**にパワーFFTのグラフ表示器と**Auto Power Spectrum**アイコンを追加します．**図20(a)**は編集後のブロックダイヤグラムで，**図20(b)**は実行結果です．このノイズ波形はサイン波形がベースになっていますので，パワーFFTにはサイン波形と同様鋭いピークがみられます．

一見ランダムで周期性がないようにみえますが，FFT解析をするとその正体がわかります．

**(3) カオス波形**

この場合も，パワーFFTのグラフ表示器と**Auto Power Spectrum**アイコンを追加します．**図21(a)**は編集後のブロックダイヤグラムで，**図21(b)**はフロントパネル(実行結果)です．パワーFFTには何もピークは見られません．ピークとなる周波数成分が現れていません．

実は，これがカオスの特徴の一つです．

〈図20(a)〉 編集後のブロックダイヤグラム(ノイズ波形の場合)

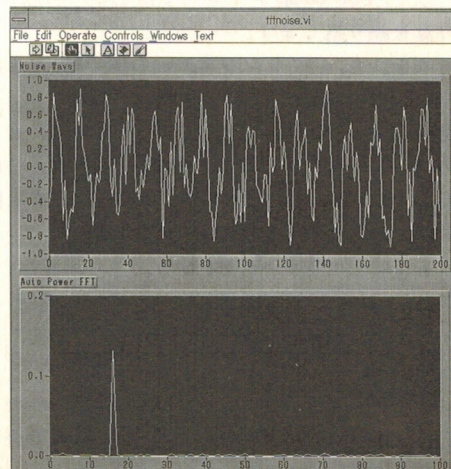
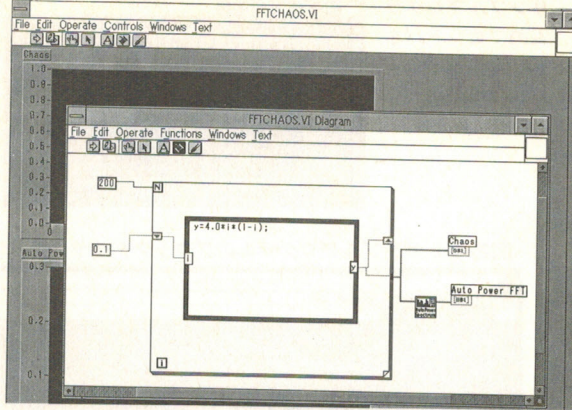
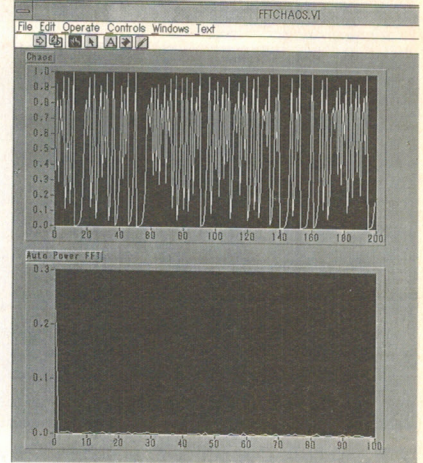
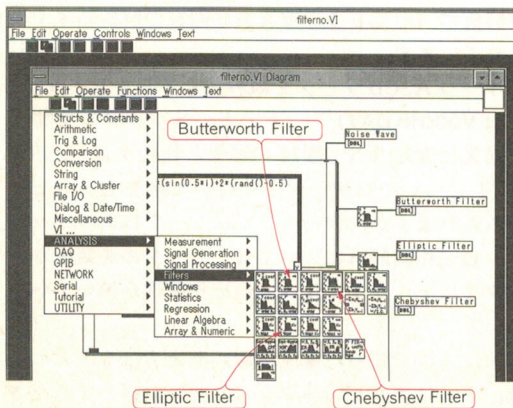
〈図20(b)〉 ノイズ波形とそのFFT

〈図21(a)〉 編集後のブロックダイヤグラム(カオス波形の場合)

〈図21(b)〉 カオス波形とそのパワーFFT

## ノイズ波形にフィルタをかける

図15(a)のノイズ波形は，ノイズが重畳した実測波形を想定したものです．実験などで経験したことがあると思います．このままでは，以後の計測・制御などの処理に困りますので，フィルタ(平滑化ともいう)をかけて“きれいな”波形にしてみましょう．

LabVIEWではブロックダイヤグラムの**Functions**の**Analysis**の中に**Filters**として各種フィルタが用意されています(図22)．ここでは，3種類のフィルタを使うことにします．**Butterworth Filter**，**Elliptic Filter**，**Chebyshev Filter**です．図15(a)のフロントパネルに3種類のフィルタに対応したグラフ表示器を追加します．この表示器にもフィルタと同名の名前を付けます．

図23は，各フィルタのアイコンとグラフ表示器の端子などを配線して編集したブロックダイヤグラムです．フロントパネルに戻って実行します．図24のようになりました．フィルタを通すことによって“きれいな”波形になりました．

〈図22〉 ブロックダイヤグラムで3種類のフィルタのアイコンを選択

## ASCIIデータを取り込む

実際に測定したASCIIデータをLabVIEWに取り込んでみましょう．そして，実測データのパワーFFTと自己相関解析(Auto Correlation)についても行ってみます．

実は，LabVIEWの中には便利なサンプルプログラムがたくさん入っています．その中で，**SMPLFILE.LIB**の中の**Temperature from File.vi**はASCIIデータを取り込んで波形表示するプログラムです．これを使うことにします．

フロントパネルとブロックダイヤグラムをそれぞれ図25に示します．フロントパネルの波形表示用のグラフ表示器(**Temperature Graph**)とASCIIデータ表示用の表示器(**String Read from File**)については，**Controls**メニューの**Graph**と**String**の中にそれぞれあり

〈図23〉 編集後のブロックダイヤグラム(ノイズ波形をフィルタに通す)

〈図24〉ノイズ波形をフィルタに通した実行例

〈図25(a)〉サンプルプログラムのフロントパネル(Temperature from File.vi)

〈図25(b)〉サンプルプログラムのブロックダイヤグラム

ます．これまで使ってきたものと同じもので，名前が **Temperature Graph** や **String Read from File** となっているだけです．

**図25**(b)のブロックダイヤグラムをみてください．上記表示器の端子のほかに，フロッピのアイコン(**Read Characters From File.vi**)と **Extract Numbers.vi** というアイコンがあります．前者は **Functions** メニューの中の **UTILITY** の中の **File** にあります(**図26**)．後者のアイコンはサンプルプログラムのメインのプログラムで，いわば LabVIEW のオリジナルです．これを利用する立場から中身については，せんさくしないことにします．

さて，これから取り込む実験データは電流波形ですので，グラフ表示器の名称を **Current waveform** に変えます．また，できるだけたくさんのデータを表示できるように ASCII データ表示器(**String Read from File**)の枠も大きく広げます．さらに，パワー FFT と自己相関解析(**Functions** メニューの中の **Analysis** の **Signal Processing** の中にある)のアイコンをブロックダイヤグラムに追加します．

**図27** は，このようにして編集したフロントパネルとブロックダイヤグラムです．これで準備が整いました．いよいよ ASCII データの取り込みです．2 種類の電流波形を取り込みます．

ここに，実験で得られた電流波形の ASCII データがあります．ASCII データの入ったフロッピをパソコンの A:ドライブに差し込んでから，フロントパネルの実行ボタンをクリックします．自動的にファイルダイアログが開き，ASCII データの一覧が表示されます(**図28**)．

**V00015.DAT**(データ 1)というファイルを選び，**OK** ボタンをクリックします．クリックと同時に，**図29** の実行結果が得られました．グラフ表示器には取り込んだ電流波形が，ASCII データ表示器にはデータ一覧がそれぞれ表示されています．また，パワー FFT と自己相関解析についてもこの電流波形固有の解析結果が表示されています．

もう一つ ASCII データを読み込んでみましょう．こんどは **V00018.DAT**(データ 2)というファイルです．実行結果は**図30** です．**図29** の結果と比較すると，パワー FFT についてはデータ 1 には少し幅をもったピークがみられますが，データ 2 にはピークはありません．また，自己相関解析については，ピークの先鋭度からデータ 2 はデータ 1 に比べて自己相関性が薄いといえます．

なお，ここで使った電流波形の測定方法などについては，トラ技コンピュータ 5 月号[(6)]を参照してください．

〈図26〉Functions メニューの UTILITY の中にある各 File のアイコン

〈図27(a)〉 ASCIIデータ(波形)表示と波形解析表示用のフロントパネル

〈図28〉 フロントパネルで実行ボタンをクリックするとFile Dialogが開く

● **計測・制御機能が豊富なLabVIEW**

ここで紹介した応用例は，LabVIEWのもつ機能のいくつかにすぎません．非常にたくさんの計測・制御に特化した機能があります．また，自社のA-D変換ボードやGPIBボードのドライバとしてLabVIEWを使うことにより，LabVIEWの特徴を最大限に引き出すことができます．相性のいいソフトとハードを組み合わせることにより，LabVIEWはより一層使いやすい計測・制御用のツールになりうると思います．

なお，LabVIEWを動かしたパソコン環境は以下のとおりです．

> 本体：PCiN V-specVD2/66(486DX2/66 MHz)
> メモリ：24 Mバイト

〈図27(b)〉 サンプルプログラムを使って編集した後のブロックダイヤグラム

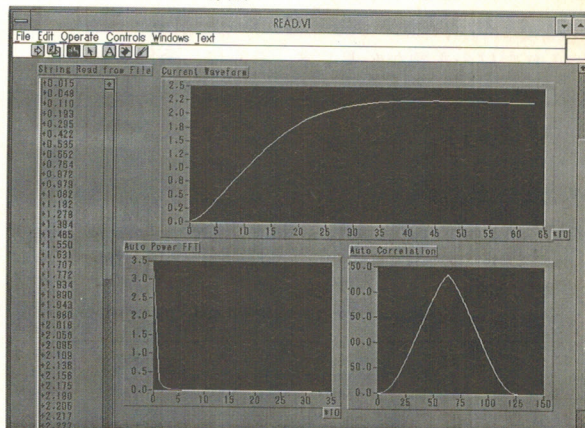
〈図29〉 ASCIIデータを取り込んで波形表示と波形解析を行う(データ1の場合)

〈図30〉 ASCIIデータを取り込んで波形表示と波形解析を行う(データ2の場合)

◆ **参考文献** ◆

(1),(2) 国内リリース元：サイバネットシステム(株)，☎03-5978-5410．

(3) 国内リリース元：住商エレクトロニクス(株)，☎03-5228-5620．

(4) 国内リリース元：日本ナショナルインスツルメンツ(株)，☎03-5472-2970．

(5) 例えば，伊東著「カオスって何だろう」ダイヤモンド社．

(6) 臼田；Originで波形解析，トラ技コンピュータ5月号 pp. 93～99，1995年．

# 第8章

## LabVIEWを活用するノウハウ(1)

# GPIBシステムで計測を自動化する方法

伊藤　敏

● **はじめに**

GPIBとはなんでしょうか．イメージをつかむために，つぎに述べるような場合を見ていきましょう．

ある物質，例えば液晶に電場をかけて，光の透過する性質の時間変化を見る場合を想定します．ここでは液晶を例にしますが，もちろんほかの物質に置き換えてもかまいません．このようなシステムを作るためには，**図1**のような系を考えます．

液晶に電源から電場をかけます．さらに光源から液晶に光をあて，透過する光の強度を検出器(フォトダイオード)で電流に変換し，さらにOPアンプで電圧に変えます．それを「メータ」で読み取ります．そして，それらを時間ごとに「ノート」に記録します．液晶試料のセッティングやその他の作業はすんでいるとします．人手による実験をする場合は，

(1) 液晶にかける電圧をセットする．
(2) 時計をスタートさせる．これは(1)と同時に行う．
(3) 時間ごとに，メータを読み取り，ノートに記録する．
(4) ノートに記録した測定データから必要な物理量を計算する．

これだけの手順が必要です．さらに，条件によっては(1)の液晶にかける電圧を変える必要がでてくるかもしれません．

(3)の時間ごとに読み取る間隔が大変短い場合は，人手による実験は不可能になります．また，人が読み取ることが可能な時間，たとえば10分間隔での測定であったとしても，この測定が30時間続ける必要があった場合は，可能ですが大変です．

筆者は昔(また昔話)，これに似た実験をしました．10分間隔で30時間測定を続ける必要がありました．測定を開始した午前中は他の人と話をしたりして，10分間隔とは暇な実験だとたかを食っていたのですが，夜になってくると食事も短時間で済ませなければならないし，10分間隔の呼び出しがいかに大変であるか感じました．

さらに，深夜からは睡魔との戦いです．このあたりからは測定システムの横に簡易ベッドをおいて8分おき(2分前)に起きるように目覚し時計をかけ，できるだけ体力を消耗しないように工夫しながら実験をしました．

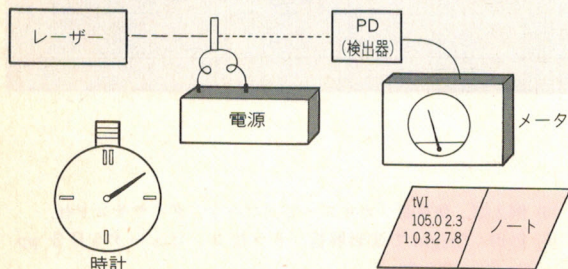
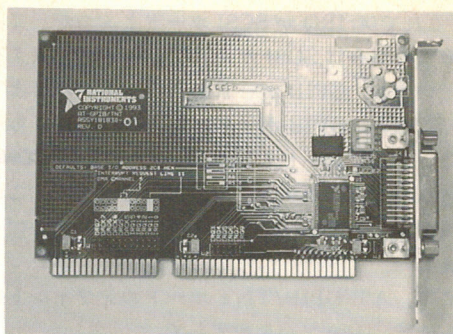
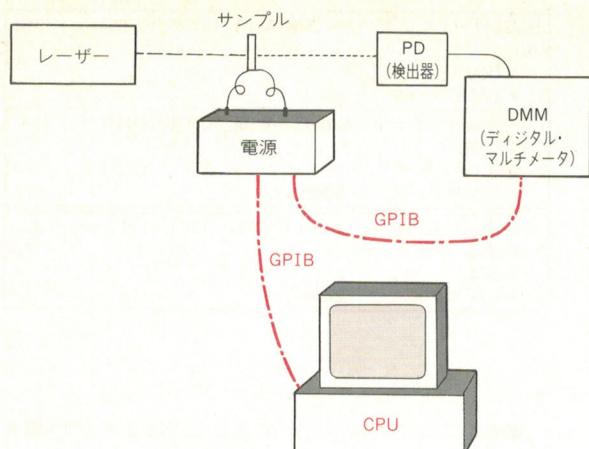
この実験はとにかくやり遂げました．しかし，何年かして，今度は1分おきに約3週間連続測定をする必要がでてきました．しかも，今度は液晶に電場ではなく磁場をかけます．さらに，その磁場を測定のたびに，すなわち1分おきに変化させます．これはもはや人間技ではありません．

計測システムを作る専門家ならば，時間ごとに磁場を変化させ(電磁石を使えば磁場は電流で制御できる)，同時に光の強度をコンピュータに記録するシステムを作り上げることができるでしょう．

しかし，筆者には何をどのようにしていいのかわかりません．しかも，コンピュータに入れるにはアナログ量をディジタル量に変換しなければなりません．A-D変換やD-A変換用のチップのマニュアルや解説書を「見ました」が，まさに眺めるだけしかできません．

さらに，コンピュータのInput/Output関係の本も見ました．そのときの気持は何とかやらなければならないのですが，このようなシステムを作るためには，勉強から始めて数ヵ月あるいは1年ぐらいの時間が必要のように思えました．

しかも，この実験が終わればこの計測システムは使わなくなります．継続して使うほどの汎用性を持っていませんから．そのようなときに，つぎに述べるようなGPIBの存在を知りました．

〈図1〉ある計測システムを見る(人手をかける場合)

〈図2〉GPIBを用いたある計測システムの例

〈写真1〉ナショナルインスツルメンツ製
AT-GPIB/TNTボード

計測器の標準インターフェースとしてヒューレットパッカード社から提案されたインターフェースバスをもとにしてできたのがGPIB(General Purpose Interface Bus)です．一般に計測システムを組む場合，コネクタの種類，ピン配置信号の論理信号レベル，出力形式などを決めて各パートを組み上げていきます．

しかし，このような計測システムを作り上げるには多くの費用と時間そして技術が必要とされます．作り上げられたシステムが汎用性をもち，大量生産をすることにより値段が下げられればそれは価値を生み出すでしょう．しかし，研究開発的な分野では，すぐに別の計測システムが必要となる場合が多くあります．

このような状況を考え，ヒューレットパッカード社が作っていたインターフェースバス(HPIB)を基礎にしてGPIBが作られたのです．GPIBは計測機を対象とした機器相互の入出力や制御，さらにコネクタも統一することを目指したインターフェースバスです．

まさにこのGPIBを使えば，短時間で必要な計測システムを構築することができると直観しました．今の場合，GPIBインターフェースをもつ光源用の電源と光強度を電圧に変換した後の電圧を読むディジタルマルチメータ，そしてコンピュータ側に付けるGPIBボードがあればいいのです．

早速，電子関係の仕事をしている人にGPIB関係の機種があるかどうか相談をしました．そして，筆者のほうでGPIBボードを用意すれば，後の計測器は借りられることがわかりました．

そこで，すぐに**図2**のようなシステムを組み立てました．手で電源を回していたのを，コンピュータからGPIBを通じてセットします．さらに，光の強度はGPIBでコントロールされたディジタル・マルチメータを通じてコンピュータに取り込まれます．もちろん，時間はコンピュータ自身が管理しています．

このように，GPIBを使用することにより，このようなシステムを短時間で構築し運用することができました．もちろん，あの悪夢のような30時間「束縛」実験のような苦労はありません．

GPIBも万能ではありません．特に速い変化を追わなければならない実験を行うには，それなりの難しさがあります．しかし，計測の素人でもそれなりのシステムを作れる点はすばらしいことです．詳しくはCQ出版㈱からでている岡村氏のGPIBまたは標準ディジタルバス(IEEE-488)関係の本を参照してください．

では実際にGPIBボードを使ってみましょう．

## GPIBボードのインストール

ここではGPIBボードとしてAT-GPIB/TNTボード(ナショナルインスツルメンツ製)を用いました．これはPC/AT互換機のATバスに差し込むタイプです(**写真1**)．

その他に，XTバスやEISAバス，PCIバスを使用するタイプも用意されています．前述のノート型パソコンのPCMCIAを利用するもの，パラレルポート(プリンタポート)やシリアルポートを利用するタイプまであります．

### ● ハードウェアのインストール

ボードを差すときは，IRQ，DMA，I/Oアドレスの設定などがあります．筆者はこれが苦手で，とにかくデフォルトのまま差し込みましたが，案の定動作しません．

筆者のマシンにはSCSIボードとイーサネットカードが差してあります．各ボードのIRQ，DMA，I/Oアドレスを**表1**に示します．

今回用いたGPIBボード，AT-GPIB/TNTのデフォルト値は，

▶IRQ………………11
▶DMA ……………5
▶I/Oアドレス……**2c0h**

です．これでは，IRQ が SCSI カードとぶつかっています．筆者の場合は SCSI カードにハードディスクをつなげています．GPIB ボードをデフォルトのままで差し込んだとき，GPIB ボードは使えないのはもちろんのこと，SCSI ボードも認識されなくなり **CONFIG.SYS** が大き過ぎると表示してそのままでした．このときは大変慌てました．久しぶりにリセットキーに手がいきました．

そこで，AT-GPIB/TNT ボードのデフォルト値を変更します．ジャンパを設定して，

▶IRQ……………… 9
▶DMA …………… 7
▶I/O アドレス…… **2c0h**（変更なし）

とします．このように変更をしてからボードを入れ直すと，無事にコンピュータが立ち上がりました．

この「すったもんだ」で半日近くロスをしてしまいました．急がばまわれとは本当によくいったものです．

● **ソフトウェアのインストール**

つぎに，ボードに標準で付属している NI-488.2 for Windows 3.1 J をインストールします．その他に DOS 用のソフトが付属します．通常の Windows 用ソフトのインストールと同じです．プログラムマネージャの File-Run から **A：¥SETUP** として，あとは質問に答えるだけです．

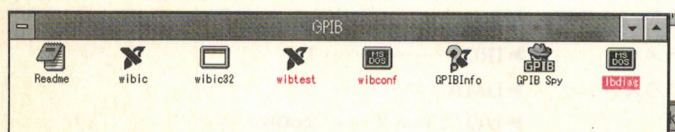
インストールが終わると，GPIB というグループができます（**図 3**）．先程述べたようにハードウェアをデフォルト値から変更していますので，ソフト上でも設定値を変更します．**wibconf** をダブルクリックして，**GPIB0** を選択します．先程設定したハードウェアの値と一致するようにセットします．

● **インストールの確認と動作チェック**

きちんとインストールできたかどうかを，ハードウェアとソフトウェアの両面からチェックをします．ここでは，ハード/ソフト両方の診断プログラムが付属していますから，これらを実行するだけでチェックできます．この段階で不都合があれば修正します．

はじめに，ハードウェア診断プログラムを実行しましょう．

GPIB ケーブルはすべてはずしてください．GPIB グループの中から **ibdiag** をダブルクリックします．I/O アドレス，DMA，IRQ の順に質問をしてきます．先程の設定値を代入してください．問題なくインストールされていれば，エラーメッセージなしで終了します．

〈図 3〉 GPIB グループ

〈表 1〉 筆者使用の各ボードの条件

| |
|---|
| SCSI カード：アダプテック製 AHA-1542CF（デフォルトで使用）<br>▶IRQ……… 11<br>▶DMA …… 5 |
| イーサネットカード：EtherLink Ⅲ 3C509 COMB（デフォルトで使用）<br>▶IRQ……………… 10<br>▶I/O アドレス…… **300h** |
| 今回用いた GPIB ボード：AT-GPIB/TNT のデフォルト値<br>▶IRQ……………… 11<br>▶DMA …………… 5<br>▶I/O アドレス…… **2c0h** |

筆者はここで，エラーが出ました．はじめ IRQ を 9 ではなく 12 にセットしてボードを差し，ここまで無事に通過してきました．しかし，ここで，IRQ 12 はどこかで使用しているとメッセージが出ました．

よく考えてみればそうですね，IRQ 12 は PS/2 マウスが使っていました．しかたなく，またボードを取り出して，ジャンパを入れ替えて上に書いたような設定（IRQ 9）に変更をし，ここをパスしました．このような診断プログラムがあるのは大変助かります．

つぎにソフトウェア診断プログラムを実行します．**wibtest** の中から「ソフトウェア診断-GPIB0 テスト」を選択します．結果を**図 4** に示します．これは，正常に終了しました．

接続してある GPIB の情報を得るには **GPIBInfo** をダブルクリックしてください．**図 5** のように情報が表示されます．

このボードを実際に使ってデータを取り込むのは LabVIEW で行いましょう．

## LabVIEW から GPIB ボードを使用する

● **GPIB 装置を接続する**

では，つぎに LabVIEW の上から GPIB ボードを制

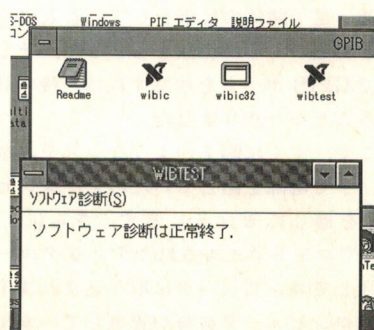
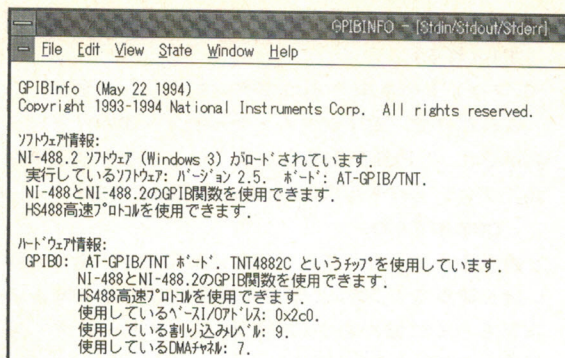
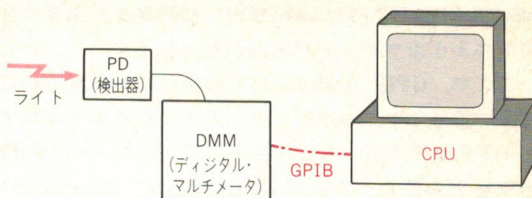
〈図 4〉 ソフトウェア診断-GPIB0 テスト

〈図5〉GPIBの情報

御し，データを取り込むことを考えましょう．LabVIEWの概要はすでに第7章で述べられていますので，参照してください．

**図6**に示すように，ディジタル・マルチメータのみを用いた非常に簡単な計測システムを考えます．光の強度をフォトダイオード-ディジタル・マルチメータを通じてコンピュータに取り込むだけです．

アドバンテスト社のディジタル・マルチメータTR6845(**写真2**)を用いてシステムを作ります．ケーブルを通してGPIBボードにつながっています．

GPIB装置を使用するには，GPIB番地(アドレス)を設定しなければなりません．各装置により番地設定法は違いますので，マニュアルを見て設定しましょう．今回の場合はソフトでアドレスをセットするようになっています．その他にディップスイッチで設定する場合があります．

〈写真2〉アドバンテストのディジタル・マルチメータTR6845の正面

〈写真3〉GPIBケーブルを重ねて接続したところ

〈図6〉光の強さをはかる

具体的には，**写真2**に見られる[Shift]キーと[GP-IB]を同時に押します．[Down]と[Up]キーにより数字を増減させ，今の場合はディジタルマルチメータのアドレスを2にセットしました．

今回はディジタルマルチメータとコンピュータをつなげるだけなのでGPIBケーブルは1本で足りますが，いくつかの装置をつなげる際には，GPIBケーブルは**写真3**のようにどんどん上に重ねてつなげること(つなげる装置の数や各ケーブルの長さ・全ケーブルの長さの和に制限がある)ができるように工夫されています．

### ● チュートリアルの利用

LabVIEWは奥の深いソフトウェアです．また，かなり専門的な機能も含まれ，使いこなすまでにはある程度の時間が必要とされます．そのために，ていねいなチュートリアルが付いてきます．これを利用してLabVIEWの基本的な操作方法をマスタできるようになっています．今回はこのチュートリアルを利用します．

LabVIEWのチュートリアルマニュアルには，GPIBボードを制御する例が含まれています．はじめにそれから紹介しましょう．

LabVIEWを起動したら，Fileメニューから

〈図7〉EXAMPLES¥GPIB¥SMPLGPIB.LIB内のLabVIEW<->GPIB.viのフロントパネル

LabVIEWディレクトリ中の**EXAMPLES¥GPIB¥SMPLGPIB.LIB**内の**LabVIEW<->GPIB.vi**を開きます.**図7**のようなフロントパネルが開かれます.

ここで,GPIB AddressはGPIBに接続されている装置の番号です.つながっているディジタル・マルチメータの装置番号は2です.そして,このメータの機能や測定条件のセットをGPIB用言語でCharacters to Writeに書き込みます.

GPIB言語は使用する機器によって変わります.マニュアルで確認をしましょう.今回のディジタル・マルチメータのGPIB制御用の言語は**表2**のようになっています.

練習のために,実験室に転がっていた抵抗の抵抗値を測ってみます.抵抗は100Ωです.測定機能は抵抗ですから**F3**,測定レンジは**R3**,サンプリング速度は**PR2**を選んでみました.手順は,

(1) GPIB Addressに装置番号2を入れる.

(2) ディジタル・マルチメータに抵抗を上の条件で測定することを指示する.

**F3R3PR2E**

を Characters to Writeの窓に書き込む.

(3) WriteとReadのスイッチをクリックして上に倒す(入れる).

(4) フロントパネルのRunボタンをクリックする.

これだけで,ディジタル・マルチメータがリモート表示され,GPIB制御命令が来ていることを知らせた後,フロントパネルのCharacters Readの窓に,

**098.87E+0**

と表示されました.これで測定は終了です.あっけないほど簡単です.特に,筆者の場合はIRQやDMAの設定をろくに確かめずにGPIBボードをセットして苦労しただけに(深く反省).

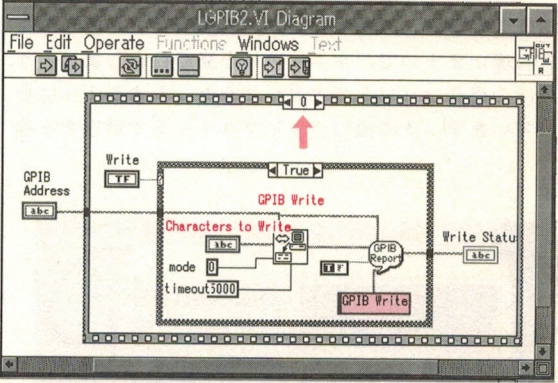
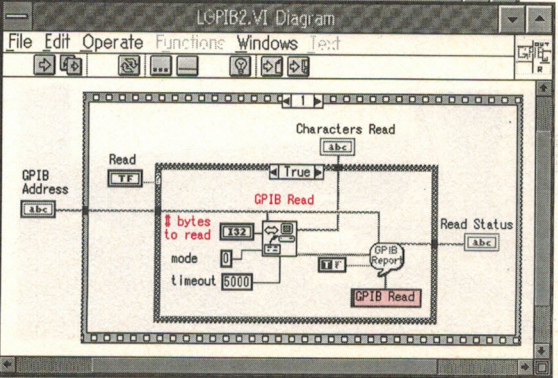
では,当初の目的である光の強度を測りましょう.ディジタル・マルチメータに光センサをつけて抵抗がいくつになるのかまったくわかりませんので,レンジは自動(AUTO)である**R0**にします.Characters to Writeに命令「**F3R0PR2E**」を入れます.Characters Readの窓に「**823.2**」と表示されて測定終了です.

では,このチュートリアルの**LabVIEW<->GPIB.vi**の中身を覗いてみましょう.フロントパネルのWin-

〈表2〉ディジタル・マルチメータTR6845用GPIB制御言語の抜粋

| | コード | 機　能 |
|---|---|---|
| 測定機能選択 | F1 | 直流電圧測定 |
| | F2 | 交流電圧測定 |
| | F3 | 抵抗測定 |
| | F4 | 抵電流抵抗測定 |
| | F5 | 直流電流測定 |
| | F6 | 交流電流測定 |
| 測定レンジ選択 | R0 | 自動 |
| | R2 | 30 mV,30 Ω,30 μA |
| | R3 | 300 mV,300 Ω,300 μA |
| | R4 | 3000 mV,3000 Ω,3000 μA |
| | R5 | 30 V,30 kΩ,30 mA |
| | R6 | 300 V,300 kΩ,300 mA |
| | R7 | 1000 V(DC),750 V(AC),3000 kΩ,3000 mA |
| | F8 | 30 MΩ,10 A |
| | R9 | 300 MΩ |
| サンプリング速度 | PR1 | FAST |
| | PR2 | MID |
| | PR3 | SLOW |
| デリミタモード | DL0 | CR/LFおよびEOI |
| | DL1 | LFのみ |
| | DL2 | EOIのみ |
| 測定の開始 | E | |
| パラメータの初期化 | Z | |

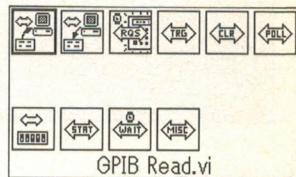
〈図10〉
従来型GPIB WriteとGPIB Read VIの機能

〈図8〉LabVIEW<->GPIB.viのブロックダイアグラム 場合1

〈図9〉LabVIEW<->GPIB.viのブロックダイアグラム 場合2

dowsメニューの中からブロックダイアグラム(**Show Diagram**)を選択すると，図8のようなブロックダイアグラムが開かれます．

大中の四角で囲まれた中に**GPIB Write**と表示されたVIが見られます．このVIからは5本の線が出ており，それぞれ，GPIB番地，GPIB命令(Character to Write)，モード，タイムアウト，GPIB Report(状態を調べるVI)につながっています．なかなか複雑そうですね．

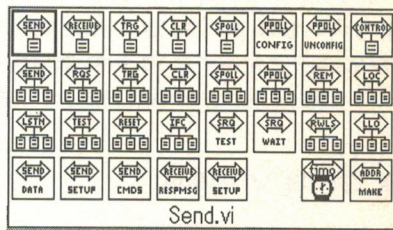
一番外側の四角の上にある「**0**」の横の矢印をクリックしましょう．図9のように画面が変わります．今度は**GPIB Read**というVIがみられます．そして，6本の線が出ています．**byte to read**という線が加わっています．そして，一番外側の四角の上が「**1**」になっています．

**GPIB Write**と**GPIB Read**と表示されているVIの機能を図10に示します．

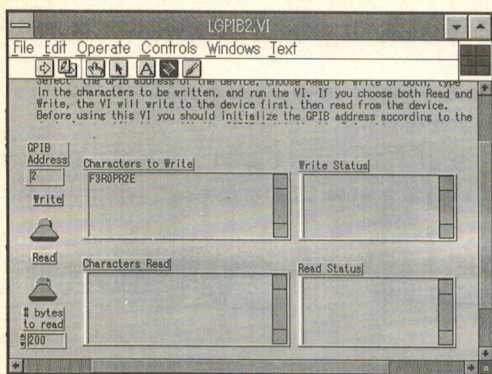
その他に，図11に示すように，従来型のGPIBに対しては全部で10のVIが用意されています．簡単に表

〈表3〉従来型GPIB VIの機能

| 名　前 | 機　能 |
|---|---|
| **GPIB Initialization** | GPIB初期化 |
| **GPIB Triger** | GPIBトリガ |
| **GPIB Clear** | GPIBクリア |
| **GPIB Serial Poll** | GPIBシリアルポール |
| **GPIB Wait** | GPIB状態待ち |
| **Wait for GPIB RQS** | GPIB SRQ待ち |
| **GPIB Status** | GPIBコントローラの状態表示 |
| **GPIB Misc** | よりハードウェアに近い処理用 |

〈図11〉
従来型GPIB VI

〈図12〉
GPIB 488.2 VI

〈表4〉GPIB IEEE 488.2 VIの機能

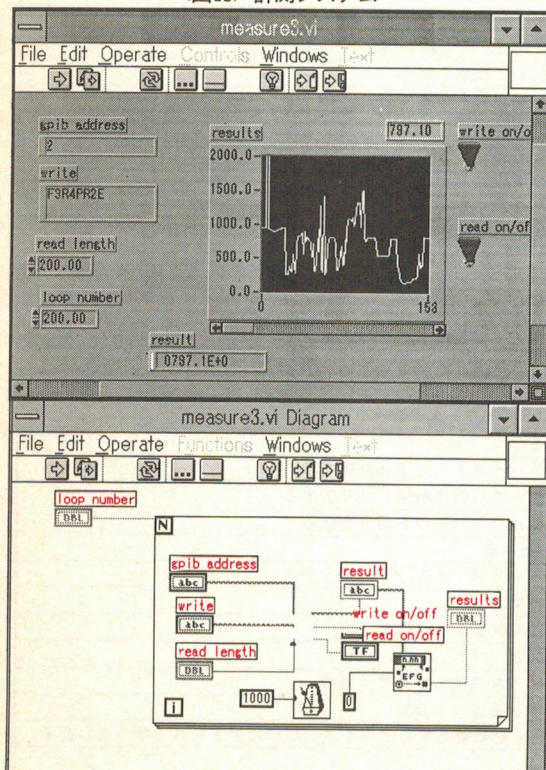
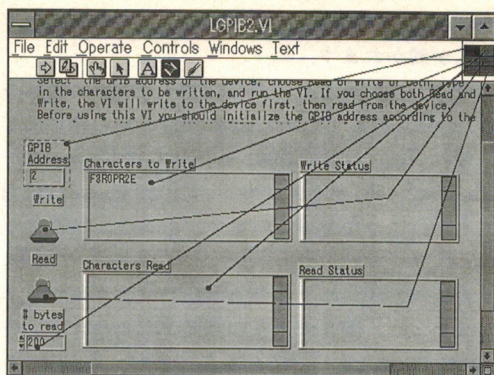
| | 名　前 | 機　能 |
|---|---|---|
| | **Send** | データをGPIBデバイスに送る |
| | **Receive** | GPIBデバイスからデータを読み取る |
| | **Trigger** | デバイスをトリガする |
| | **DevClear** | デバイスをクリアする |
| | **ReadStatus** | デバイスをシリアルポールして状態を得る |
| | **PPollConfig** | デバイスをパラレルポールに参加させる |
| | **PassControll** | 制御資格を別のデバイスに受け渡す |
| | **SendList** | 複数のデバイスにデータを送る |
| | **FindRQS** | どのデバイスがサービスを要求しているか判断 |
| | **TriggerList** | 複数のデバイスを同時にトリガする |
| | **AllSPoll** | すべてのデバイスをシリアルポールする |
| | **PPoll** | パラレルポールを実行 |
| | **PollUnconfig** | パラレルポール参加構成デバイスの構成を解除 |
| | **DevClearList** | 複数のデバイスを同時にクリアする |
| | **EnableLocal** | 複数のデバイスのローカルモードをイネーブルする |
| | **EnableRemote** | 複数のデバイスのリモート・プログラムモードをイネーブルする |
| | **FindLstn** | すべてのリスナを発見 |
| | **TestSys** | 複数のデバイスにIEEE488.2自己テスト命令を出す |
| | **WaitSRQ** | サービス要求まで待機する |
| | **ResetSys** | バス，メッセージ交換，デバイスすべてを初期化する |
| | **SendIFC** | IFC(Interface Clear)を使ってGPIB機能をクリアする |
| | **SendLLO** | すべてのデバイスにローカルロックアウト命令を送る |
| | **TestSRQ** | SRQ線の現在の状態を判断する |
| | **SetRWLS** | デバイスをローカルアウトのリモートモードに設定する |
| ローレベル入出力VI | **RcvRespMsg** | デバイスからデータを読み取る |
| | **ReceiveSetup** | GPIBボードがデータを読み取るように設定 |
| | **SenCmds** | GPIBコマンドバイトの送信 |
| | **SendDataBytes** | 直前にアドレスされたデバイスにデータバイト送信 |
| | **SendSetup** | デバイスがデータを受信できるように設定 |
| ジェネラルVI | **MakeAddr** | 1次アドレスと2次アドレスを必要とするデバイスのためにアドレスを組み合わせる |
| | **SetTimeOut** | タイムアウト時間の変更 |

〈図13〉 コネクタを追加する

3にまとめました．

LabVIEWでは従来から使われたきたGPIBだけでなく，1987年に制定された新しいGPIB規格IEEE488.2に対応するVIも31個用意されています．**図12**と**表4**にそれらのVIと持っている機能を示します．

## ◆ GPIB VIの利用

チュートリアルに付属する**GPIB VI**を利用して，とにかくGPIB計測装置を動かすことができました．これからはいかに使うかです．

〈図15〉 計測システム

〈図14〉 GPIB VIをSub VIにする

しかし，前項で実行したのはフロントパネルのRunボタンをクリックすると1回データを読み込むということだけです．これだけでは，長時間の測定をGPIBに任せておくことはできません．繰り返しデータを読み込む必要があります．

そこで，**GPIB VIをSub VIとして利用して，繰り返し測定に使えるようにします**．ここで，**Sub VI**とはプログラム言語でいうとサブルーチンに相当します．さらに，得られたデータをグラフに表示できるようにしましょう．

### ● GPIB VIをSub VIとして使う

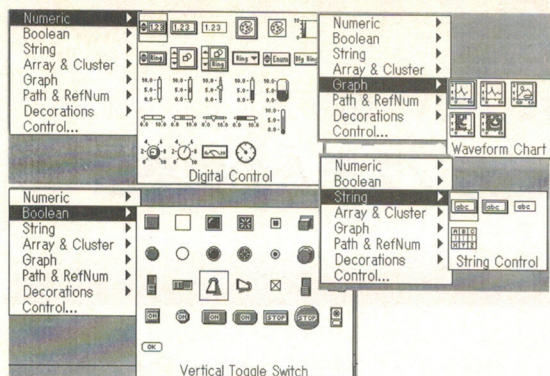
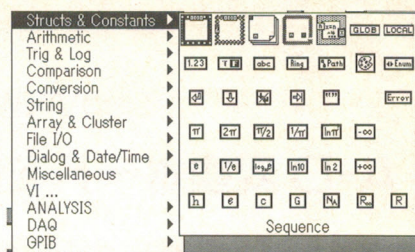
もう一度，**図7**のフロントパネルを見てください．外から入れなければならないデータと，外へ出すデータの数を数えると，

> GPIBアドレス，GPIBに書き込むデータ，GPIBから得られるデータ，WriteとReadを「真」(True)にするスイッチ，読み込むバイト数

の六つです．この六つをコネクタとして登録しましょう．**図13**のフロントパネルから編集モードボタン(鉛筆の形)をクリックして編集モードにしてください．

アイコンベース(フロントパネルの右上)をポップアップして，メニューから**Show Connector**を選択します．コネクタ端子のパターンを選択します．今回は，六つの入出力があります．6端子のパターンを選びました．各端子に**GPIB VI**の各機能を割り当てます．

コネクタの左上の端子をマウス左ボタンでクリックして端子を黒色にします．**GPIB VI**内の**GPIB Address**をクリックします．すると，破線で取り囲まれます．フロントパネルの空白領域でマウスの左ボタンをクリックすると破線は消え，コネクタの黒色をしていた左上の部分は灰色になります．これで，コネクタの左上の端子は**GBPI Address**に割り当てられました．

〈図16〉 Numeric, Boolean, String, Graph

〈図17〉 Structure & Constants

同じ操作で残り五つの機能をコネクタの各端子に割り当てます．各機能と割り当てられたコネクタの対応を**図14**に示します．

割り当てがすんだら**Sub VI**として利用が可能になります．名前を**GPIB-2.VI**にして保存しておきましょう．

● **計測システムを作る**

おおげさな題ですが，前項で作った**Sub VI**を使って，200回繰り返して光の強度を計測するシステムを作りましょう．用いるハードウェアは前と同じです．

**図15**のように**gpib address, write, result**をフロントパネルの**String**メニューから取り出して張り付けます．同じように，**read length, loop number**を**Numeric**から，**write on/off, read on/off**を**Boolean**から，**results**(グラフ)を**Graph**から選びます．**図15**のように各条件を入れ，ダイアログパネルで**for**ループを書き足して各要素を配線すると計測システムをコントロールするVIのできあがりです．

**図15**は光センサーの前に灯を置いたり，手で覆いをしたりして光の強度を変化させたものを1秒おきに記録し，グラフ化したものです．後は，これを適当に自分のシステムに合わせて変えれば，立派なシステムのできあがりです．

**図15**を作るのに必要なメニューを**図16, 図17**に示します．

ダイアログパネルのメニューの中に「電球」のマークがあります．これをクリックして実行させると電球が点灯し，データの「流れ」がダイアグラムパネル上に小さな丸印で表示されます．作ったシステムのデバッグの際には有効でしょう．

最後に，今回のGPIB計測システムに用いたコンピュータは，Dell 433L(CPU 486DX)メモリ16Mバイトでした．

◆ **参考文献** ◆

(1) 岡村廸夫；標準ディジタル・バス(IEEE-488)とその応用，CQ出版㈱．

## デモバージョンの提供について

特集の中で紹介した，ナショナルインスツルメンツ社，インテリジェントインストルメンテーション社，日本ヒューレット・パッカード社から，デモセットを提供いただきました．ご希望の方はアンケートハガキのモニタープレゼントの欄にご希望のデモセット番号[(a)～(e)]をご記入ください．ただし，提供品は各々数点ですので，希望者が多い場合は抽選とさせていただきます．

<編集部>

### ◆ナショナルインスツルメンツ社

(a) 媒体；1.2バイト(PC98用)3.5インチフロッピ，内容；同社のハードウェア選択支援アドバイザ．

(b) 媒体；1.44バイト(Windows用)3.5インチフロッピ，内容；Windows用LabWindows/CVIのデモ．日本語ガイド付き．

(c) 媒体；1.44バイト(Windows用)3.5インチフロッピ，内容；Windows用LabVIEWのデモ．日本語ガイド付き．

### ◆インテリジェントインストルメンテーション社

(d) 媒体；CD-ROM，内容；Visual Designerのデモバージョンで，DIAGRAM.EXEとRUN.EXEから構成されており，設計に使うDIAGRAM.EXEは製品と同じ．RUN.EXEは 実行時間が1分に制限されている．デモ版で作成されたブロック図は製品版でも動作する．

### ◆日本ヒューレット・パッカード社

(e) 媒体；1.44バイト(Windows用)3.5インチフロッピ，内容；HP VEE for Evaluation Kit．製品版との違いは，(1) インストール後，60日経つと起動できなくなる．(2) 計測パネルの数がかぎられている．(3) プログラムの保存名がEVAL.VEEのみ．簡単なガイドが付属する．詳細なマニュアルはオンラインヘルプを参照．HP-IB計測機をコントロールするためにはHP-IBインターフェースが必要．

# 第9章

## LabVIEWを活用するノウハウ(2)
# A-D変換ボードを使った測定システムの構築法

井上祥史

### ● はじめに

例えば，家庭の照明・温度・安全などを，コンピュータを使って集中管理することを考えてみます．そのためには，明るさ・室内温度・火災・出入口の開閉などの情報を継続的に取り込んで監視し，設定値と比較することによって，調光したりヒートポンプのスイッチを入れたり，また必要に応じて警報を出さなければなりません．

このようなシステムには，同時に多くの信号を取り込み，それらのデータを処理・解析して，判断した結果を出力できるコンピュータとセンサとの間を仲介をする機能をもつ回路が必要です．これを実現するのがアナログ-ディジタル入出力ボードで，高速のA-D/D-A変換器といくつかのタイミング回路，そしてバッファを備えています．

多くのアナログ信号をコンピュータに取り入れるために普通取られる方法は，マルチプレクサと呼ばれる切り替えスイッチを介して一つのA-Dコンバータに取り込むという，見かけ上の多チャネル化の方法です．A-D変換器のサンプリング時間やマルチプレクサのチャネル間の切り替え時間に依存して，計測できる範囲が決まります．

一方，出力のディジタル信号は8ビットまたは16ビットの信号をいくつかのポートから，D-A変換してアナログ出力をしますが，いずれも電流容量は小さなものです．これらアナログやディジタルの信号の入出力を行うには，ボードから見れば相手のコンピュータの機種に依存する環境設定とドライバが必要です．

〈写真1〉I/Oボードの装着状態
(およそ35 cm×10 cmの大きさで，ケースの幅いっぱいのスペースを必要とする)

ここでは，ナショナルインスツルメンツ製のI/Oボードを用い，LabVIEW上でコントロールする手順と一例を紹介します．

## 入出力ボードのインストール

### ● 入出力ボードの概要

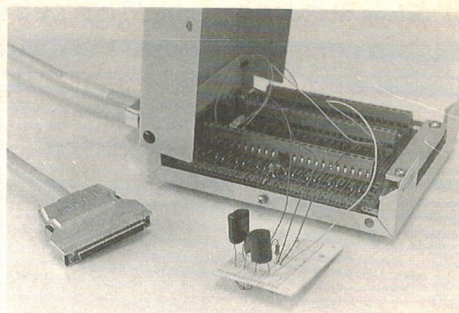
使用する入出力ボードはナショナルインスツルメンツ製のAT-MIO-16XE-50で，ボード上にはスイッチやジャンパ線がなく，必要なすべての設定をソフトウェアで行うことができます．

このボードをAT互換機に装着した様子を**写真1**に示します．またこのI/Oボードの概要について，**表1**に示します．

これらを上回る機能が必要な場合には，より高機能なボード(例えばアナログ入力で64チャネル，サンプ

〈表1〉アナログ/ディジタル入出力ボードAT-MIO-16XE-50の概要

| | | |
|---|---|---|
| アナログ入力 | チャネル/分解能 | 16チャネル(差動入力で8)/16ビット |
| | サンプリング速度 | 20kサンプリング/秒(最大) |
| | ゲイン/入力範囲 | 1, 2, 10, 100(プログラムコントロール)/−10V～+10V |
| | バッファ | 512サンプル(FIFO) |
| アナログ出力 | チャネル/分解能 | 2チャネル/12ビット |
| | 変換速度 | 20kサンプリング/秒(最大) |
| | 出力範囲 | ±10V(0.1Ω出力インピーダンス，±5mA) |
| ディジタルI/O | チャネル | 8 input/output(3ステート) |
| タイミングI/O | チャネル | 2 up/down(24ビット)，1スカラー(4ビット) |
| | クロック | 20MHz(カウンタ)，100kHz(タイマ) |
| | トリガ | 16種コントロール |

〈写真2(a)〉 68ピン用端子盤
(テスト回路を接続したところ．ドライバ1本で簡単に接続できる)

〈写真2(b)〉 別の角度から見た様子

リング速度500Kサンプリング/秒のAT-MIO-64E-3など)も用意されています．

● **インストール手順**

LabVIEWがインストールされていれば，入出力(I/O)ボードをそのまま使用することができます．最新のドライバやサンプルを追加してLabVIEWに組み込むために，NI-DAQと呼ばれる添付ソフト(3.5インチディスク5枚)をプログラムマネージャからインストールします．この他に，データ解析用のソフト(3.5インチディスク4枚)もインストールしておきます．

LabVIEWのインストール時には，データ収録ドライバやハードウェア環境設定ユーティリティが **¥LABVIEW** の中に置かれます．これらをLabVIEWのアイコンと同時に表示させるようにしておきます．

I/Oボードをセットして LabVIEWを実際に使い始めるときに，このユーティリティを使って次のように環境設定を行います．

(1) **WDAQCONF** アイコンをクリックする

(2) デバイス1の構成/テストをクリックする

デバイスEシリーズボード/AT-MIO-16XE-50

▶DMA ……………6

▶IRQ………………5

▶ベースアドレス…**220H**

(3) 保存をクリック

“スイッチレスデータ収録デバイスのソフトウェア構成が成功しました”が出ると，ハードウェアテストを行うことができます．デバイスの構成で，デフォルト値(DMA=6，IRQ=10)を使うとイーサネットカード(著者ではIRQ=10)，SCSIカード(IRQ=11)やPS/2マウス(IRQ=12)などがある場合には，ハングアップしたりエラーを出してしまいます．その場合は，上の(2)に戻り，適切なパラメータの割り当てを捜します．

(4) テストをクリック

▶極性/範囲 ………−10V〜+10V

▶モード……………デファレンシャル(差動入力)

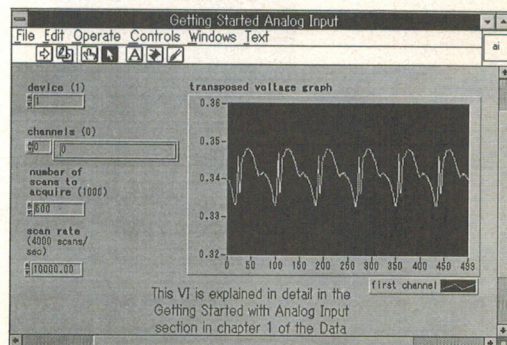
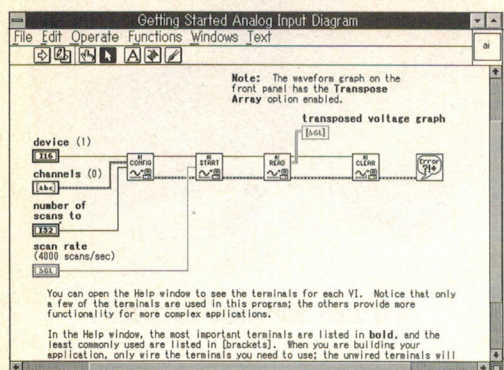
▶AMUX-64T ……なし(外部マルチプレクサなし)

にしておいて，アナログ入力，アナログ出力，ディジタルテスト，カウンタテストを順次行い，いずれもError=0が出れば機能テストは完了します．

● **実測テスト**

I/Oボードの診断結果に問題がなければ，さっそくLabVIEWのアイコンをクリックして立ち上げます．Fileメニューの中のopenから，**LABVIEW/EXAMPLES/DAQ/RUN_ME.LLB/Getting Started Analog Input.VI** を読み込みます．このパネル画面にあるデバイス1はAT-MIO-16XE-50のボードのことで，**WDAQCONF** のところで指定したとおりです．チャネルは0としましょう．

何も接続していないときに−10.00Vの直線がグラフ表示されたら，ソフト/ハードとも動いています．デフォルトでは取り込み速度4000ポイント/秒のうち

〈図1〉 Getting Started Analog Input.VIで観測した蛍光灯の強度変化

〈図 2〉 Getting Started Analog Input.VI のダイヤグラム

1000 ポイント分が表示されていますので，仮想装置(VI)のブラウン管のフルスケールの時間幅が 1000/4000＝0.25 秒となります．10 分割されているブラウン管を想定するならば 25 ms/div になります．

次に，適当な信号源をチャネル 0 に接続して，波形を表示させてみましょう．AT-MIO-16XE-50 の入出力端子数は 68 で，これに合うメス型コネクタは市販品からはなかなか手に入りません．結線をするには**写真 2** に示すようなネジ止めできる端子盤(SCB-68：ケーブル付き)が便利です．

信号源には高感度のフォトダイオード(S3072)を用いました．このダイオードはほとんどソーラーセルとなり，単独で電圧源とすることができました．

**WDAQCONF** でおこなった設定は差動入力ですから，ACH0(チャネル 0)と ACH8(チャネル 8)の68 番と 34 番のピンにダイオードの両端を接続します．これらの入力は AIGND を基準とするため，順方向電圧を測定するには AIGND(67 番)と ACH8 を接続します．

先ほどの何も接続しないときのテストで－10.00 V が表示されたのは，差動入力の基準点が不確定のためにマイナス方向に OP アンプが飽和したためです．

蛍光灯の下で **Getting Started Analog Input.VI** を **run** させてみると，**図 1** のような 120 Hz の光起電力が観測されます(愛知県なので 60 Hz×2)．Y 軸方向のスケールの自動調整には波形グラフにパレットを追加しておくと便利です．**図 1** ではスケールを手動で変えました．

取り込み速度(スキャンレート)をデフォルトの 4000 ポイントから 10000 ポイントに増やし表示域を 500 ポイントにすれば，同じ 0.25 秒のフルスケールで，より精度の高いグラフが得られます．

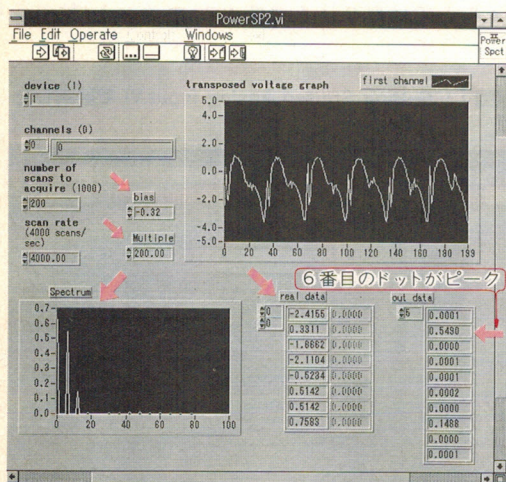
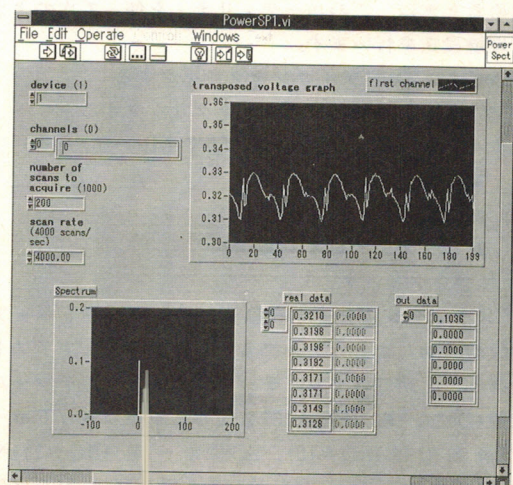
**図 2** に **Getting Started Analog Input.VI** のダイヤグラムを示しました．このダイヤグラムでわかるとおり，I/O ボードの環境設定をして，バッファや取り込み速度(scan rate)のパラメータを読み込み，測定を開始するという一連のタスクの流れが，利用者の思考どおりに組み立てられている様子がよくわかります．

## 自作 VI の作成

### ● パワースペクトルの表示

**LABVIEW/EXAMPLES** にはアナログ in，アナログ out，アナログ in/out，ディジタル I/O などのたくさんのサンプルがあります．これらのサンプルは，階層化されて用意されている VI〔Easy，中間(intermediate)，基本(advanced)〕のうち，中間の VI を多く使った構成になっています．この中間 VI でほとんどの入出力に必要な機能が実現できますから，高度な要求以外には基本的な **avdanced.VI** に触らなくてもよい仕組みになっています．

したがって，ここでは **Getting Started Analog**

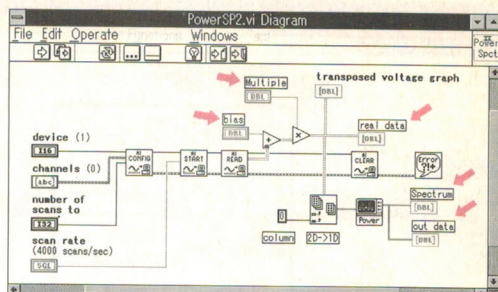
〈図 3〉 パワースペクトル表示 VI のフロントパネル

〈図 4〉 直流成分が大きく表示されたパワースペクトル

**Input.VI** にパワースペクトルの分析用 VI とその表示器を付け加えて，自分用の周波数分析器 **PowerSP2.VI** を作ってみます．

まず，**Getting Started Analog Input.VI** のパネルに，パワースペクトル表示器とバイアスおよび増幅器の数値入力器，そして波形配列データ(2 次元)の数値表示器とパワースペクトル・データの配列(1 次元)の数値表示器を取り付けてみます(**図 3**)．

バイアス(bias)および増幅器(Multiple)を取り付ける理由は，ハード的に適切な電圧にしなかったため，入力信号の中の 0.3〜0.4 V の直流成分をバイアス電圧として引算し，それを増幅してパワースペクトルを大きく表示するためです．もしバイアスを設けないと，直流成分がパワースペクトルの原点に大きく現れてスペクトル波形が見にくくなるでしょう．**図 4** にバイアスや増幅器を設けないで，直流成分のみが大きく表示されたパワースペクトルの例を示します．

これらの表示器，入力器をパネルに取り付けた後，ダイヤグラム画面を**図 5** のように修正します．**AI Read.VI** からの 2 次元配列データから負のバイアスデータを足し算し，**Multiple** で増幅しています．配列に定数を掛けたり足したりすれば，配列の各要素は定数倍されたり定数が加えられたりします．この計算結果を波形グラフ(ラベルは **transposed voltage graph**)と数値表示器(ラベルは **real data**)に出力しています．

一方この 2 次元配列データを左側の要素列だけの 1 次元配列に直すのが **Index Array.VI**(Array & Cluster の中にある)です．**Index Array.VI** はメニューから引き出した直後では 1 次元用ですから，アイコンの右下スミをマウスで引き延ばして 2 次元にします．そして，第 0 列のみを抽出するために，行指定を無効(任意の長さの縦方向の配列が読める)にするように，右下の二つの端子の上のほう(行)をポップアップして **Disable Indexing** を選びます．そして，定数の **0** を下の端

〈図 5〉パワースペクトル表示 VI のダイヤグラム

子(列)に接続します．

こうすることで任意の長さの 2 次元配列が，第 0 列要素からなる任意の長さの 1 次元配列に変換されます．これを **Power Spectrum.VI** に入れて表示したのが**図 3** のグラフだったのです．

このようにして得られたパワースペクトルの数値表示器を見ると，6 点目(表示器の一番上が 5 になっている)で最も大きい値 0.5490 をもっています．入力の速さ(scan rate)は 4000 点/秒で，このうち 200 点を表示しましたから，フルスケールは 200/4000＝0.05 秒です．このフルスケールに対応する周波数は 20 Hz となり，これがパワースペクトルの分解能(一つの点の表す周波数幅)になります．

したがって，この 6 点目のピークは 120 Hz を表しています．同じように，次のピーク(0.1488)の 12 点目は 240 Hz を表します．これは，前述の Mathematica を使ったフーリエ変換の結果と一致しています．

## ● 過渡現象のための入出力

アナログ-ディジタル I/O ボードでは，信号を取り込むと同時に出力信号を出すことができます．過渡現象や計測制御を行う系では，出力を電圧源として電流を観測したり，センサで計測をしながら制御信号を出力することになり，入出力を同時に行うことは最も一

〈図 6〉過渡現象測定の接続回路

〈図 7〉過渡現象測定 VI のフロントパネル

〈図 8〉 過渡現象測定 VI のダイヤグラム

般的な使い方です．

アナログ出力の電流容量は±5 mA と小さいので，外部に駆動回路を接続します．しかし，ここでは試験的にアナログ出力端子を電圧源としてわずかな電圧を加え，制限電流の範囲内で過渡現象を調べてみます．

回路の接続は**図 6** のようにしました．アナログ入出力のグラウンド(AIGND と AOGND)は共通になっています．*RLC* 回路に加える電圧は矩形波で，電圧の大きさや時間幅は可変できるようにします．

それでは，さっそく VI の作成に入りましょう．入出力を同時に行いますから，I/O ボードのバッファを介さないで 1 点ごと入力波形をタイムチャートに記録し，1 点ごとアナログ出力をします．ある程度高速の現象を捕らえる場合には，誤差が蓄積してこないハードウェアタイミングが適当と考えられます．そして，用意されているアナログ入力のサンプル VI にアナログ出力を付け加えるように考えていきます．そのため，**LABVIEW/EXAMPLES/DAQ/ANALOGIN/ANALOGIN.LLB/Cont Acq&Chart(hw timed).VI** を選びました．

パネルウィンドウに出力チャネル(out channel；文字配列)，出力電圧，矩形波のスタート点，終了点の入力器を取り付けます(**図 7**)．印可電圧と電流波形の両方を見るので，入力チャネルには 0：1 とセットします．入出力とも同じボードですからデバイスは共通で 1 となります．ダイヤグラムは**図 8** のようになります．

1 点ごとにアナログ出力するため，**AO Write One Update.VI**(中間 VI；**intermediate VI**)を付け加えます．この VI の入力は 1 次元配列の電圧データですから，**While** ループの繰り返し回数 **i** が **start<i<end** の条件を満たすときに 0.5 V を出力するような **Case** 構造にし，**Bild Array** を使って 1 次元配列に直してから **AO Write One Update.VI** に接続します．

セットしたパネルの値で **RUN** すると，**図 7** の表示器に見られるように，0.5 V の矩形波が 30 と 70 点の間で発生します．そしてストップボタンが押されるまでチャートは流れ続けます．矩形波の 30 点目の立ち上がりと 70 点目の立ち下がりでの過渡波形は，予想どおり逆特性となりました．必要であれば，解析的に得られた式で計算した厳密解を表示して，実測値と比較し測定誤差を評価・検討することもできます．

この例では単純な矩形波を発生させましたが，フォミュラーノードなどを使えば，任意の入力波形に応答する過渡現象を見ることができます．作成した VI を端子を持った SubVI として登録することは容易です．

モジュールとして SubVI を作成する場合は，実行状態として保存する必要があることはいうまでもないでしょう．任意のアイコンパターンをカラーで編集するのは楽しいものです．

### ● おわりに

以上見てきましたように，I/O ボードを LabVIEW 上でコントロールし，計測制御系を組み上げていくことは比較的簡単です．特にドライバやジャンパスイッチなどの環境設定に費やす時間を省けることは，より多くの機能を付加して質の高い制御系を構成する機会を与えてくれます．

また，豊富な解析 VI を用いて，ほかの装置からのデータの解析ツールとして LabVIEW を利用することもできます．LabVIEW がマルチタスクとして実行されていれば，連続した現象を計測，解析しながら任意の深さの SubVI をモニターしたり別の解析ツールを走らせたり，あるいはネットワークで接続してデータを転送したりコントロールすることが可能です．

これからより多くの分野で LabVIEW の VI の資料集や使用例が蓄積，報告されて行くことになるでしょう．

# 第10章

## 画像データをパソコンで扱うためのノウハウ
# ビデオキャプチャ・ボードを活用した計測法

伊藤　敏

ここでは，カノープス製のVIDEOPAQ-ATを使い，ビデオ信号をコンピュータに取り込みました．そのときの使用例を紹介します．機材はビデオ信号源としてフロッピカメラ(キヤノン RC-250)を使いました．

## ビデオキャプチャ・ボードについて

### ● カードの外観

**写真1**に示すように，少し大きなカードです．パッケージには，このカード以外にシンクロ転送ケーブルとドライバやユーティリティソフトの入ったフロッピ，User's Guideが付属しています．

その他に必要とされるのはPC本体とDOSです．すなわちMS-Windowsは必要としません．しかし，実際のキャプチャのところでも述べますが，MS-Windows上のDOSモードからの画像取り込みも可能です．

ボードのセットアップに必要な項目として，I/Oアドレスの設定INT(割り込み)の設定があります．いずれもPCにつけてあるほかのボードと重複しないようにします．筆者はデフォルトの設定で用いました．

また，ボード上には，ビデオ信号入力ソースを選択するスライドスイッチがあります．これによって，コンポジット端子とS端子が選択できます．

### ● カードの装着

筆者のマシンにはCPUにぶつかりカードが差せないスロットがありました．

カノープス製のPHOTOPAQ表示ボードを持っている場合は付属のシンクロ転送ケーブルで表示ボードをつなぎます．今回は違う表示ボードを使いましたのでケーブルによる接続はしません．

### ● ソフトのインストールと実際の使用

つぎはソフトのインストールです．付属のフロッピをセットして**SETUP.EXE**を実行すれば終わりです．

まず使ってみましょう．

(1) DOS上で使用する

**SETUP**でインストールしたディレクトリに移って各ドライバを組み込みます．

**VIDEOPAQ**(↵)
**GRAPH**(↵)

表示ボードとしてPHOTOPAQを用いる場合はGRAPHの代わりに，

**PHOTOPAQ**(↵)

とします．つぎにビデオ画像取り込みユーティリティを起動します．

**VP**(↵)

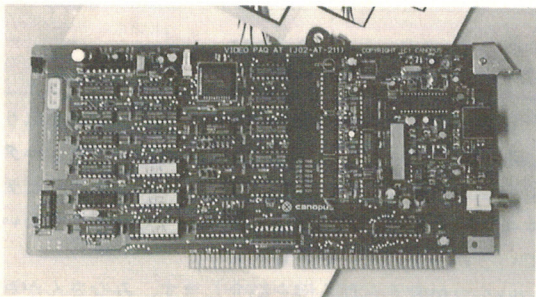
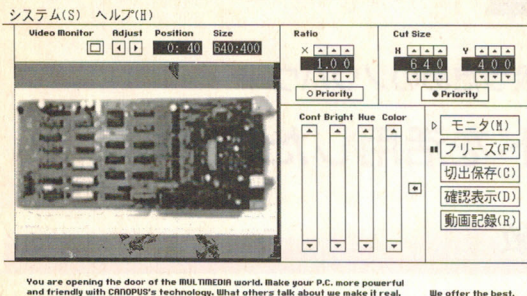
後は，画面の指示通りにすれば，静止画像でも動画でも取り込みが可能です．付属の**VP.EXE**は**表1**の画像ファイル形式に対応しています．

(2) MS-Windows上で使用する

DOSの上で(1)と同じようにMS-Windowsを起動する前に，各ドライバ**VIDEOPAQ**と**GRAPH**を組み込んでおきます．Windowsが立ち上がったらDOSの窓を開けて，DOSの場合と同じく，ビデオ画像取り込みユーティリティを起動します．

〈写真1〉VIDEOPAQ-ATの外観

〈表1〉VIDEOPAQ-ATの対応する静止画像フォーマット

| 拡張子 | 対応する画像ファイル形式 |
|---|---|
| JPG | JPEG形式 |
| RGB | RGB形式 |
| PIC | RGBライン形式 |
| P | RGBブロック形式 |
| TIF | TIFF形式 |
| BMP | DIB形式(MS-Windows)<br>▶フルカラー　▶8色<br>▶256色　▶モノクロ<br>▶16色 |

〈図 1〉 画像取り込み中の様子

〈写真 2〉 名古屋城

**VP**(↵)

ただし，DOS を全画面表示にしないと取り込み中の画像が表示されず，うまく動作しませんでした(**図 1**).

## 画像を取り込む

では，実際に画像を取り込んだ例を示しましょう．**写真 2** は名古屋城です．取り込んだ画像は**表 1** のフォーマットで保存できます．画像そのものを保存するのでしたら，ファイルサイズの小さい JPEG で保存したほうがよいでしょう．

ここでは，後でこれらの画像を「画像処理」することを考え，**BMP** で保存しました．また，**BMP** で保存すると自分の好きな画像を特別のツールなしで MS-Windows の壁紙として利用できます．これは気分転換として有効です．いらいらするときは気持ちの落ちつく山や海を，気分が乗っているときはそれなりの画像をはりつけてはどうでしょうか．

少し工夫をすると，面白い画像が取り込めます．ビデオ画像を少し視点(両目の間隔分)を変えて撮り，並べるとステレオ図ができます．

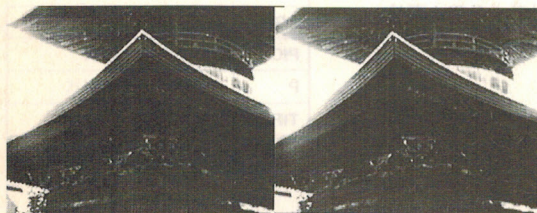
詳しい原理は，拙著の「Lotus 1-2-3 による理工系シミュレーション入門」を見ていただきたいのですが，右の図を右目で左の図を左目で見るようにする(遠くの風景を見るように目の焦点を合わせると見やすい)と，並んだ二つの図が合わさり，立体的に見えます．

〈写真 3〉 多宝塔のステレオ図

**写真 3** に名古屋の最古の建物と言われている多宝塔を撮った例を示します．真中の塔のひさしが手前に浮き出てきています．少し訓練をすれば見えるようになりますので，ぜひ試してください．

今回，筆者はフロッピカメラを用いましたが，もちろんビデオカメラの画像も取り込みが可能です．ビデオに録画した中から良いと思われる画像を取り込み，最近入手が容易になったカラープリンタを使って印刷をすればまさに，立体スキャナになります．

ただし，良い点ばかりではありません．画像の質が PhotoCD などと比べると落ちます．ただこの点は今回試用したカノープスのビデオキャプチャ・ボードの問題ではなく，ビデオ信号を取り込むボードの限界です．CCD 素子で画像を電気信号に変えた後ビデオ信号に変換するときに画質が落ちてしまうようです．

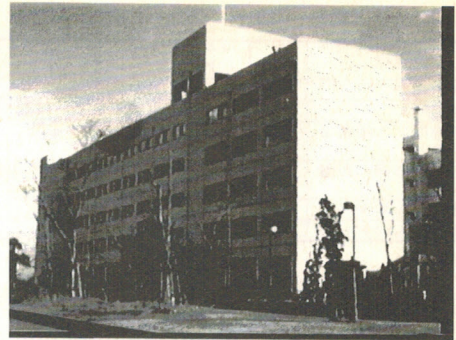
しかし，ディジタルカメラはまだ値段がかなり高く，現状で手軽に画像をディジタル化するにはこの方法か，さきほどの PhotoCD を用いるのが手軽です．

ビデオキャプチャは映像を撮ってから数分で静止画像・動画像共にコンピュータに取り込めます．家族や友人達と楽しみながらできる点ではほかにはない良い点でしょう．最近のデータベースソフトは画像を入れることができます．簡単に画像を取り込める点を活かすと，ユニークな画像データベースを作ることができるでしょう．

## 応用例

ビデオキャプチャ・ボードでいろいろな画像を取り込み楽しむだけならば，わざわざトラ技コンピュータで取り上げるほどのことはありません．しかし，ビデオキャプチャ・ボードは技術者にとっても工夫しだいでいろいろな応用が考えられます．

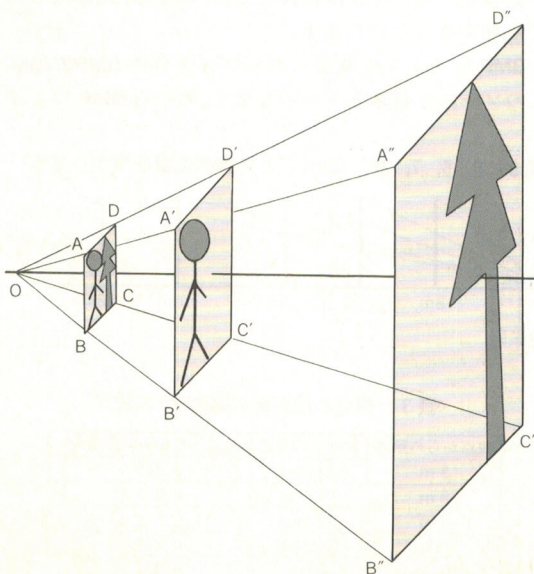
いくつか考えられる例を紹介します．みなさんが利用されるきっかけになれば幸いです．

〈写真4(a)〉 建物 ①

〈写真4(b)〉 建物 ②

### ● 建物などの大きさを測る

**写真4**を見てください．同じような写真が2枚並んでいます．しかし，写っている建物の大きさが少し違いますね．これは同じ建物を建物までの距離を10m変えて撮影した写真です．これだけのデータから写真に映っている建物の高さを推定するのが目的です．三角形の相似を使って考えてみましょう．

少しごちゃごちゃとしますが我慢してください．**図2**はO点を視点としてA" B" C" D"面にある木とA' B' C' D'面にいる人を投影面であるABCD面に写す様子を表したものです．遠くにある大きな木が近くにいる人と同じ大きさに見えます．すなわち，3次元図形を2次元に写し込むには視点より遠いものは小さくなるように座標値を変換します．**写真4**を撮った状態を横からながめた略図を**図3**に示します．

ABが建物を表しています．この高さを未知数$x$と表しておきます．C点とF点で建物ABを撮影しています．C点,F点が視点になります．BC間の距離を，未知数$y$とします．C点とF点の間の距離$p$は，先ほどの10m移動して撮影したその距離に対応します．

ここでCEおよびFHは視点と投影面の距離です．今の場合ですと，ビデオカメラの焦点距離に相当します．$q$と表し，この値は後で決めます．DEおよびGHは撮影された画像の中の建物の高さになります．それぞれ，$r$, $s$とします．整理しますと**表2**のようになります．

三角形ABCとDECは相似です．さらに，三角形ABFとGHFは相似です．これらから連立方程式を立てると，

$$x/y=r/q \quad \cdots\cdots(1)$$

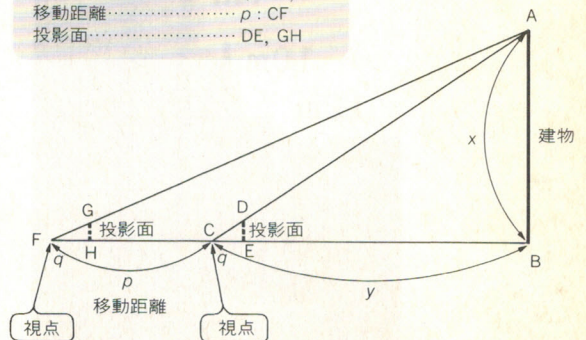
$$x/(y+p)=s/q \quad \cdots\cdots(2)$$

となります．これから$x$, $y$を求めます．

最近は計算するのが極端にめんどうになり，このような簡単な計算でもMathematicaを使うようになりました．結果は**図4**のようになります．

$$x=\frac{prs}{q(r-s)} \quad \cdots\cdots(3)$$

〈図2〉 モデル図

〈図3〉 建物の高さを推測する

〈表2〉測定点とパラメータ

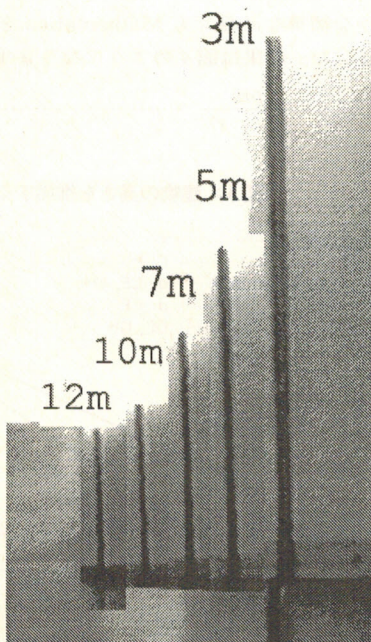
| 測定点 | 図3 | パラメータ | 内　容 |
|---|---|---|---|
| 建物の高さ | AB | $x$ | 未知数 |
| 撮影場所1 | BC | $y$ | 未知数あるいは既知 |
| 撮影場所1と場所2の距離 | CF | $p$ | 既知 |
| 視点と投影面の距離 | CD FG | $q$ | 既知(測定により決定) |
| 画像1からのデータ | DE | $r$ | 測定 |
| 画像2からのデータ | GH | $s$ | 測定 |

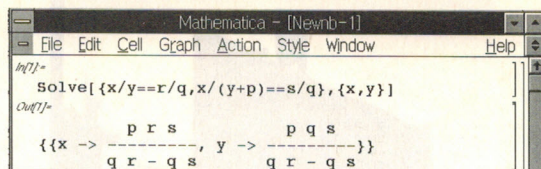
$$y=\frac{ps}{r-s} \quad \cdots\cdots (4)$$

つぎに，視点と投影面の距離 $q$ を決めます．この値は使うビデオカメラ・取り込みの解像度・用いるディスプレイによって異なると思います．長さ1mの棒を距離を変えて撮影しました．その様子を**図5**と**写真5**に示します．

**写真5**の結果から1mの棒の取り込んだ画像の画面上の長さを計った結果を**表3**に示します．この表より視点から7m以上はなれていれば $q$ はほぼ一定値になります．ここでは7mから12mの平均値0.264mをとります．

**写真4**とさらに10m離れたところから撮った写真から建物の高さを画面上から長さを求め，実際の建物の高さを計算すると20m, 17.7m, 15.4mという結果が得られました．実際の高さは17.8mと考えられますから，このような簡単な方法で求めたにしては良くあっていると思います．実際，画面上の長さを1mm読み間違うと高さにして5mくらいずれます．

〈写真5〉3，5，7，10，12mの位置から1mの棒を写した

〈図4〉Mathematicaで計算しているところ

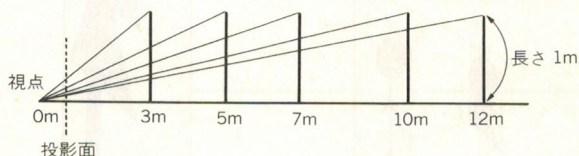
```
Mathematica - [Newnb-1]
File Edit Cell Graph Action Style Window Help
In[1]:=
  Solve[{x/y==r/q,x/(y+p)==s/q},{x,y}]
Out[1]=
              p r s               p q s
  {{x -> ---------, y -> ---------}}
           q r - q s         q r - q s
```

画面上の長さを物差しで計る場合は0.5mmが読みとりの限度です．画像のピクセルの位置を求めることで，上の値をもう少し正確に求めることが可能です．MS-Windowsに付属しているペイントブラシではメニューの［表示］の［マウスポインタの座標を表示］を設定することによりマウスのある位置をx,y表示することができます．筆者の実験した系(ディスプレイ14インチ)の画面では100mmの長さは約400ピクセルに当たりました．ピクセル単位で求める方法をぜひ試してください．

● **Mathematicaで処理する**

画像を取り込んだのですから，Mathematicaで表示してみましょう．Mathematicaで**BMP**フォーマットファイルの画像を表示するだけなら，そのままで表示できます(標準で**BMP**ファイルの表示機能をもっている)．しかし，拙著「パソコンによる数式処理ソフト活用法」(CQ出版社)でも述べましたように，数値化して表示することにより，Mathematicaを用いた処理内容の設定が容易になります．

また，Mathematicaではバイト単位でファイルを読み込めますから，**BMP**ファイルをそのまま読み込み，内部で数値化して処理することが可能です．しかし，ここでは，**BMP**ファイルをC言語を用いたプログラムで数値化してファイルに落とし，それをMathematicaで処理することにします．

**BMP**ファイルを数値化するプログラム**BMP2NUM.C**のソースを**リスト1**に示します．用いる**BMP**ファイ

〈図5〉3，5，7，10，12mごとの棒の位置を横から見る

〈表3〉視点と投影面の距離(q)の決定

| 視点からの距離 | 画面上の長さ | 視点と投影面の距離($q$) |
|---|---|---|
| 3 m | 83 mm | 0.249 m |
| 5 m | 51 mm | 0.255 m |
| 7 m | 37 mm | 0.259 m |
| 10 m | 26 mm | 0.270 m |
| 12 m | 22 m | 0.264 m |

〈リスト1〉BMPファイルを数値化する(**BMP2NUM.C**)

```
#include "stdio.h"
#include "fcntl.h"

#define BUFSIZE 1024

#define UINT   unsigned int
#define DWORD  unsigned long
#define LONG   unsigned long
#define WORD   unsigned int
#define BYTE   unsigned short

  /*  BMPファイル  ヘッダ部分の定義  */
struct tagBITMAPFILEHEADER
{
    UINT    bfType;             /*    "BM" */
    DWORD   bfSize;             /*    ファイルサイズ  */
    UINT    bfReserved1;        /*    0  */
    UINT    bfReserved2;        /*    0  */
    DWORD   bfOffBits;          /*    ビットマップイメージ  */
} BITMAPFILEHEADER;

struct tagBITMAPINFOHEADER
{
    DWORD   biSize;              /*  構造体のサイズ  */
    LONG    biWidth;             /*  幅     */
    LONG    biHeight;            /*  高さ   */
    WORD    biPlanes;            /*   1   */
    WORD    biBitCount;          /*  カラービット数   */
    DWORD   biCompression;       /*  圧縮方式（圧縮無しは0）   */
    DWORD   biSizeImage;         /*  ビットマップのサイズ（圧縮時のみ有効） */
    LONG    biXPelsPerMeter;     /*  1メートルあたりの横のピクセル数  */
    LONG    biYPelsPerMeter;     /*  1メートルあたりの縦のピクセル数  */
    DWORD   biClrUsed;           /*  使われている色の数   */
    DWORD   biClrImportant;      /*  使われている重要な色の数   */
} BITMAPINFOHEADER;

struct tagRGBQUAD
{
    BYTE    rgbBlue;            /*   青    */
    BYTE    rgbGreen;           /*   緑    */
    BYTE    rgbRed;             /*   赤    */
    BYTE    rgbReserved;        /*   0     */
} RGBQUAD;

power(x, y)
int    x;
WORD   y;
{
    int i, p;

    p = 1;
    for (i=1; i<= y; i++)
        p *= x;
    return(p);
}


main(argc, argv)
int argc;
char *argv[];
{
  unsigned int k, i, fd, nbyt;
  LONG line, bufsize, readoneline, calcpara;
  WORD    numberofpalette;
  unsigned char buf[BUFSIZE];
  unsigned char temp;
  unsigned char nishin[9];
                /* BMPファイルを開く */
  if ((fd = open(*++argv, O_RDONLY | O_BINARY)) == -1){

      fprintf(stderr, " can't open");
      exit(-1);
  }

/*  read data BITMAP Header  */
      read(fd, buf,  2);
         /*  開いたファイルがBMPファイルかどうかの判定 */
  if ( !( (buf[0] == 'B' ) && (buf[1] == 'M' ) ) ){
      fprintf(stderr, " This File is not BitMapFile.");
      exit(-1);
  }

      read(fd, &BITMAPFILEHEADER.bfSize,      4);  /* ファイルサイズ */
      read(fd, &BITMAPFILEHEADER.bfReserved1, 2);
      read(fd, &BITMAPFILEHEADER.bfReserved2, 2);
      read(fd, &BITMAPFILEHEADER.bfOffBits,   4);

/*  read BITMAP Information Header  */
      read(fd, &BITMAPINFOHEADER.biSize,          4);
      read(fd, &BITMAPINFOHEADER.biWidth,         4);  /* 画像の幅   */
      read(fd, &BITMAPINFOHEADER.biHeight,        4);  /* 画像の高さ */
      read(fd, &BITMAPINFOHEADER.biPlanes,        2);
      read(fd, &BITMAPINFOHEADER.biBitCount,      2);  /* カラービット数 */
      read(fd, &BITMAPINFOHEADER.biCompression,   4);
      read(fd, &BITMAPINFOHEADER.biSizeImage,     4);
      read(fd, &BITMAPINFOHEADER.biXPelsPerMeter, 4);
      read(fd, &BITMAPINFOHEADER.biYPelsPerMeter, 4);
      read(fd, &BITMAPINFOHEADER.biClrUsed,       4);  /* 使われている色数 */
      read(fd, &BITMAPINFOHEADER.biClrImportant,  4);

/*  read RGBQUAD * Paret Number  */
      numberofpalette = power(2, BITMAPINFOHEADER.biBitCount) ;

      for (k=0; k < numberofpalette ; k++){
          read(fd, &RGBQUAD.rgbBlue,     1);          /* パレット */
          read(fd, &RGBQUAD.rgbGreen,    1);
          read(fd, &RGBQUAD.rgbRed,      1);
          read(fd, &RGBQUAD.rgbReserved, 1);
      }

      printf("%lu ", BITMAPINFOHEADER.biWidth);      /*  画像の幅   表示 */
      printf(" %lu \n", BITMAPINFOHEADER.biHeight);  /*  画像の高さ 表示 */

  line = BITMAPINFOHEADER.biHeight - 1;
  if (BITMAPINFOHEADER.biBitCount == 1 )  calcpara = 8;
  if (BITMAPINFOHEADER.biBitCount == 4 )  calcpara = 2;
  if (BITMAPINFOHEADER.biBitCount == 8 )  calcpara = 1;

  readoneline = (BITMAPINFOHEADER.biWidth + calcpara - 1) / calcpara;

  if ( readoneline <= 4 )
      bufsize = 4;
   else
      { if ((readoneline /4 )*4 == readoneline)
           bufsize = readoneline;
        else
           bufsize = (readoneline /4 + 1 )*4;}

  while ((nbyt = read(fd, buf, bufsize )) != 0) {  /* 1行分のデータを    */
   /*  in the case of 1 bit  */                    /* bufに読み込む      */
         if (BITMAPINFOHEADER.biBitCount == 1 ){
                 for (k= 0; (k < readoneline ) ; k++ ){
                   temp = buf[k];
                     for (i = 7 ; i = 0 ; i--){
                       nishin[i] = temp % 2;
                       temp = temp /2;    }
                   for ( i = 0 ; i < 8 ; i++)
                     printf("%2u ",nishin[i]);    }
                                                    }

           else {

      for (k= 0; (k < readoneline ) ; k++)   /* 1行分の有効データの出力 */
       {    printf("%3u ",buf[k]);
                                                }

}
     printf("\n");
     line = line - 1;
  }

  close(fd);      /* BMPファイルを閉じる */
}
```

〈リスト2〉
名古屋城のデータ(一部)

BM / ファイルサイズ 264728 / 幅 616 / 高さ 428 / カラービット数 8

```
42 4D 18 0A 04 00 00 00-00 00 36 04 00 00 28 00   BM........6...(.
00 00 67 02 00 00 AC 01-00 00 01 00 08 00 00 00   ..g...ｬ.........
00 00 00 00 00 00 12 0B-00 00 12 0B 00 00 00 00   ................
00 00 00 00 00 00 00 00-00 00 01 01 01 00 02 02   ................
02 00 03 03 03 00 04 04-04 00 05 05 05 00 06 06   ................
06 00 07 07 07 00 08 08-08 00 09 09 09 00 0A 0A   ................
0A 00 0B 0B 0B 00 0C 0C-0C 00 0D 0D 0D 00 0E 0E   ................
0E 00 0F 0F 0F 00 10 10-10 00 11 11 11 00 12 12   ................
12 00 13 13 13 00 14 14-14 00 15 15 15 00 16 16   ................
16 00 17 17 17 00 18 18-18 00 19 19 19 00 1A 1A   ................
1A 00 1B 1B 1B 00 1C 1C-1C 00 1D 1D 1D 00 1E 1E   ................
中略
FA 00 FB FB FB 00 FC FC-FC 00 FD FD FD 00 FE FE   ................
FE 00 FF FF FF 00 55 46-2F 42 2F 42 2F 2F 2F 2F   ......UF/B/B////
1A 2F 11 1A 23 17 1F 1D-17 17 17 23 17 1A 1F 1A   ./..#......#....
17 23 23 1A 23 17 17 11-11 17 11 23 17 11 1A 2F   .##.#......#.../
28 28 28 2F 2F 28 2F 2F-28 2F 2F 2F 2F 28 2F 28   (((//(//(////(/(
2F 28 2F 3D 2F 28 2F 2F-2F 2F 2F 2F 42 2F 42 2F   /(/=/(//////B/B/
46 55 55 55 55 68 55 55-55 55 55 68 55 68 55 55   FUUUUhUUUUUhUhUU
70 68 68 68 68 68 70 84-70 70 70 70 84 84 84      phhhhhp a pppp ﾃ ﾃ
84 95 84 95 95 84 95 84-95 84 84 9A 95 84 95 9A   符封....ﾃ嚼..噪
84 95 95 9A 84 A9 A9 A9-95 A9 A9 A9 95 A9 A9 84   封噪ｩｩｩ鮒ｩｩ鮒ｩ.
A9 9A A9 A9 A9 9A A9 A9-A9 A9 A9 9A 95 9A A9 9A   .圸ｩｩ圸ｩｩｩｩ嚼圸圸
A9 84 A9 9A 95 9A 84 A9-84 95 84 9A 95 84 9A 84   ..嚼噪ｩ....符噪
95 9A 95 9A 84 9A 95 9A-A9 84 A9 84 84 84 84 84   伏伏..伏ｩ..ﾃﾃ.
以下略
```

パレット

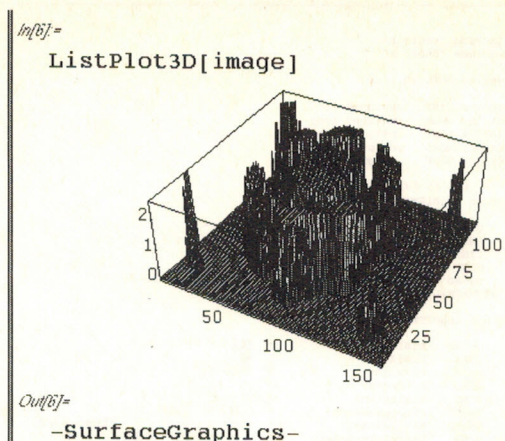
画像データのはじまり

〈図 6〉 数値データを Mathematica で表示

ルは，もちろんビデオキャプチャ・ボードから取り込んだものです．

しかし，技術的な用途ということから，フルカラーとか 256 色で取り込んだ画像をグレースケールに変換しています．**リスト 1** を見ていただければわかるかもしれませんが，簡単に **BMP** ファイルのフォーマットを**リスト 2** に示しました．**リスト 2** は**写真 3**（名古屋城）をグレースケールに変換したものをダンプしたものの一部です．

このファイルを **BMP2NUM.EXE** を使って数値に変換すると，616×428 の行列になります．これだけのデータを Mathematica で表示処理するには多量のメモリが必要です．筆者のところでは 16 M バイトの主メモリを載せていますが，200×200 でも処理にかなりの時間がかかりました．

**図 6** の例は 158×116 です．これは，ある液晶にレーザ光をあて，それの光回折像を撮影して取り込んだ **BMP** ファイルを，上記の方法で処理して表示させたものです．用いた液晶はある周期（繰り返し）構造をもっており，それの周期が回折像を作ります．この像のスポットの間隔とスポットの光強度から得られる情報より，回折像の元になった液晶の構造を調べることができます．

〈写真 6〉 人物の写真

〈図 7〉 3 次元表示

スポットの光強度を明確に表示するために，3 次元表示（**ListPlot3D[ ]**）を**図 7** に示しました．中心付近は光強度が飽和して平になっていますが，周辺の部分は場所による光強度の違いが明確にわかります．これから先の Mathematica の使い方については，拙著の「パソコンによる数式処理ソフト活用法」やその他の本を参照してください．

このような光回折や X 線回折画像の光強度処理は，昔（ほんの少し前）は写真に撮って，その写真ネガから黒化度を測る高価な専用スキャナを用いて数値化をして処理をしていました．それと比べるとビデオキャプチャ・ボードを用いるこの方法は実に簡単です．

### ● 面積比を求めよう

つぎに，面積比を求めてみましょう．**写真 6** は人物を撮影したものです．これの人物の部分の面積比を求めてみましょう．左の画像は取り込んだ画像そのものです．右は左の画像を白黒 2 色の画像に変換したものです．先で述べた Mathematica を用いる方法もあるとは思いますが，ここでは白黒 2 色に変換をした画像を処理しましょう．

ついでと言っては何ですが，面白い画像フォーマットを紹介しましょう．XPixMap ファイル（**XPM**）です．UNIX の世界では知られているフォーマットです．この特徴は，データが ASCII コードで記述されていることです．

これにより，普通のエディタで編集が可能です．ま

〈リスト3〉
XPM ファイルの一部

```
/* XPM */
static char* xpm = {
"199 310 2 2",
" 0 c #ffffff",
" 1 c #000000",
" 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0",
" 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0",
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0",
中略
" 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0",
0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1
1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0",
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0",
} ;
```



た，モノクロ，カラー，グレースケールに対応しています．例を**リスト3**に示します．Cのソースに似ていますね．

では，**写真6**をXPMフォーマットに変換しましょう．DOSやMS-Windowsの世界でXPMに対応したソフトがあるかどうか知りませんので，同じPCに載せてあるLinux(PC-UNIXの一つ)で変換します．X Window上で動作する**xv**というソフトを用います．

**xv**は**図8**に示すように，多くの画像フォーマットに対応し，また相互に変換が可能です．ここでMS-WindowsのBMPファイルを読み込み，白黒(Black & White)画像に変えて，XPMフォーマットでセーブします．その結果の一部が**リスト3**です．

ここで，ファイルのヘッダ情報のカラーテーブル(白黒だから2色しかないが)とピクセル情報部分を白は0に黒を1に，エディタの置換機能で置き換えます．そのほかに，処理をするのに余分なところを削除します．

このファイルをLotus 1-2-3に数値データとして読み込んで処理をしましょう．読み込んだら，あとは，数値データの範囲で「和/全セルの数」を求めます．白の部分は0，黒の部分は1の値をとりますから，これで，黒の部分の面積比が得られたことになります．結果は0.19という値が得られました．

今回は単純な面積比を求めるということで紹介をしましたが，面積そのものを求めることや，その他のことも，みなさん方の工夫しだいでかなりの応用ができると思います．いろいろと試みてください．

〈図8〉xvで表示したXPMファイル

## ● まとめ

カノープス製のビデオキャプチャ・ボードを用いて，画像を取り込み，いろいろと使うことを紹介しました．現在，ワープロでもビデオ画像が簡単に取り込めるようになってきています．ビデオ画像を文章に張り付けるなどの利用方法ならば，ビデオ取り込み機能の付いたワープロのほうがコスト的にも優位です．

しかし，それから一歩先を行く，すなわち取り込んだ画像をいろいろ処理していくためには，やはり今回紹介したようなビデオキャプチャ・ボードが必要でしょう．そして，それを使う人間のアイディアがもっとも大切です．

# 第11章

## 画像データをMathematicaを用いて処理するノウハウ

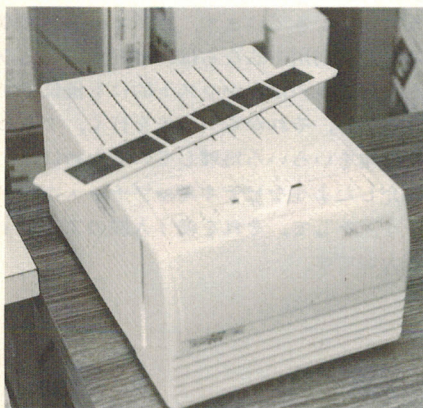
# フィルムスキャナを活用した計測法

伊藤　敏

### ● はじめに

先に，ビデオキャプチャ・ボードを使って，手軽に画像をコンピュータに取り込む方法を紹介しました．その他に画像をディジタル化してコンピュータに取り込む方法としては，ディジタルカメラやスキャナを利用する方法があります．写真にした画像をコンピュータに取り込む範囲まで拡大すると，PhotoCDとフィルムスキャナを活用することができます．

この章ではフィルムスキャナを活用する紹介をします．フィルムスキャナは筆者がたまたま利用できる立場にあったことと，仕事で液晶の画像を解析する必要があり，その経験を紹介できればと考えたからです．そして，解析したい画像をフィルムの形で入手さえすれば手軽に低解像度から高解像度までのディジタル画像が手にできます．

今回用いたスキャナは，データリンク社のScanMarker 35T(TWAIN対応機器)です．PC/AT互換機のXTバス(8ビット)用の本機専用のSCSIインターフェースカードが付属しています．このスキャナはMS-WindowsのPhotoShopの中からコントロールをして画像を取り込むことが可能です．ほかにニコンからクールスキャンというフィルムスキャナが発売されています．これはアダプテック社の154X SCSIボードをインターフェースとして利用でき，拡張スロットが不足の人には都合が良い製品です．

ScanMarker 35Tを**写真1**に示します．フィルムは1枚ずつスライドにマウントされたものと，6枚つづりのマウントされていないもののいずれにも対応できます．また，ネガフィルムとポジフィルムのどちらでも取り込みが可能です．もっとも，ネガ・ポジの反転は画像処理ソフトのほうで対応できますが．

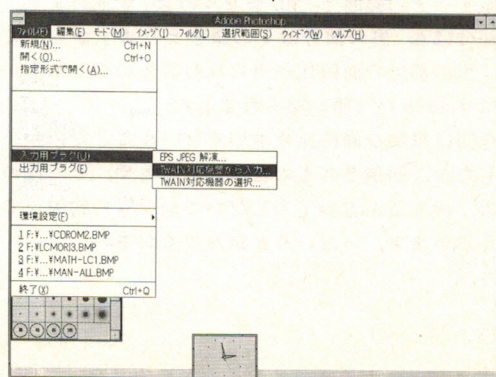
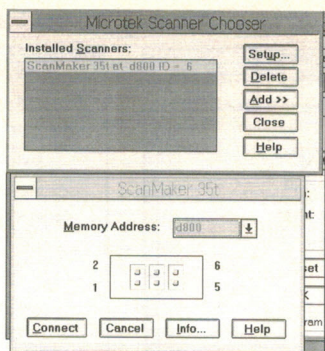
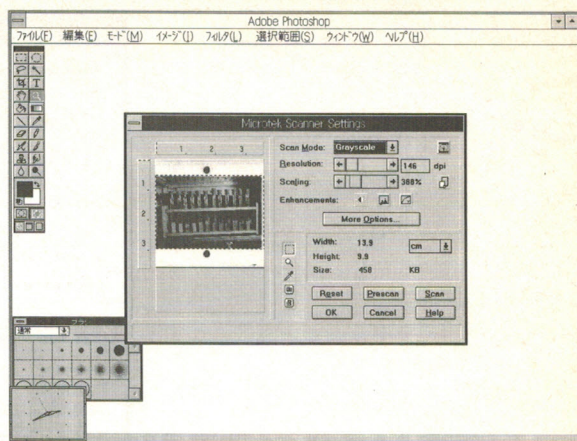
## インストール

ハードウェアのインストールは，ボードをコンピュータに差し込み，スキャナ本体とケーブルでつなぐだけです．XTバスがなかったのでATバスに差し込みました．IRQは標準では使用しません．しかし，専用SCSIインターフェースボードの標準の設定ではドライバの使用するアドレスがMS-Windows 3.1Jの使用する領域と重なっています．マニュアルの指定通りにボード上のジャンパを変更をしてから差し込みます．

つぎに，ソフトウェアをインストールします．ドライバ関係のインストールの前にPhotoShopを入れなければなりません．標準の方法でインストールします(プログラムマネージャのメニューから**PSSETUP.EXE**を実行する)．この製品にはPhotoShopが標準で付いてきました．

続いて，Scan Moduleのインストールです．ファイ

〈写真1〉ScanMarker 35T

〈図1〉PhotoShop上での設定

〈図2〉
ドライバの設定

〈図3〉取り込み条件設定

ルマネージャから **SMSETUP.EXE** を実行します．Scan Module にはディスプレイとプリンタの調整をするためのキャリブレータがあります．必要に応じて実行してください．

PhotoShop 上でスキャナ用のドライバの設定をします．PhotoShop を立ち上げたら**図1**のように，File メニューの中から「入力用プラグ」「TWAIN 対応機器から入力」を選択して，**図2**のようにスキャナドライバのインストールをします．さらにジャンパの設定をデフォルトから，ハードウェアでの設定に合わせます．

## 画像取り込みの実際

では実際に使ってみましょう．

ここでは，液晶の顕微鏡写真を用いました．フィルムをスキャナにセットして，設置の所で述べた**図1**を実行すると，すでにドライバがインストールされてますから，**図3**のような取り込み画面になります．

ここで，『Microtek Scanner Settings』の取り込み条件を設定して，『Scan』ボタンを押すとスキャナが動きだします．

〈図4〉画像取り込み中

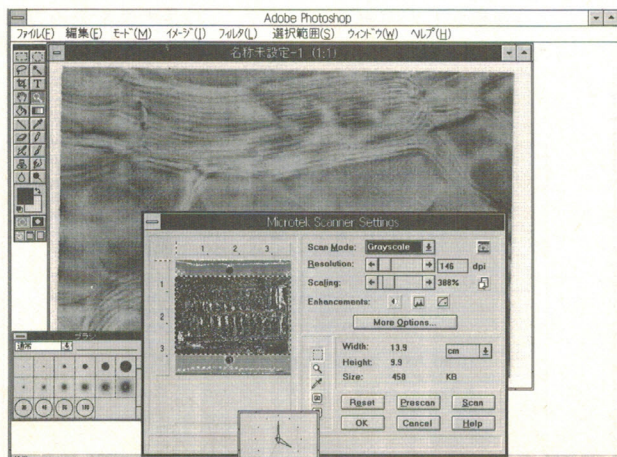



**図4**のような画像が表示されます．ここで，『Microtek Scanner Settings』の『OK』ボタンを押すと画像取り込み終了です．後はファイル名を指定してセーブすれば終わりです．

雰囲気はおわかりいただけたでしょうか？ 取り込んだ画像から大きさや長さを求める場合は，取り込んだ解像度と倍率(scaling)をきちんとメモしておきましょう．より精密な画像が欲しい場合は解像度を上げることで実現できます．

ただ，解像度を上げすぎるとファイルサイズがとんでもなく大きくなります．後の加工のことも考えて適切な解像度で取り込みましょう．

## 取り込んだ画像の処理

取り込んだ画像を解析する例を紹介しましょう．

**図4**に示す液晶の顕微鏡写真でみられた縞模様の縞の間隔を測定します．測定したい画面の一部を拡大して MS-Windows に標準で付いてくるペイントツールに読み込みます．ここで，ペイントツールのマウス設

〈図5〉ペイントツールで画像を解析

〈図 6〉距離の求め方

〈図 7〉Mathematica で計算式を求める

(a)

```
Mathematica - [Analy-1]
File Edit Cell Graph Action Style Window Help

Solve[{y==a x+b,y==-x/a+c},{x,y}]

{{x -> -(((y1 - y2) (x2 y1 - x1 y2)) /
          2                   2     2
     (-x1  + 2 x1 x2 - x2  - y1  +
                    2
        2 y1 y2 - y2 )) -
    ((-x1 + x2) (-(x1 x3) + x2 x3 -
        y1 y3 + y2 y3)) /
        2                 2     2
    (-x1  + 2 x1 x2 - x2  - y1  +
                 2
      2 y1 y2 - y2 ),
  y -> -(((-x1 + x2) (x2 y1 - x1 y2)) /
          2                  2     2
     (-x1  + 2 x1 x2 - x2  - y1  +
                    2
        2 y1 y2 - y2 )) -
    ((-y1 + y2) (-(x1 x3) + x2 x3 -
        y1 y3 + y2 y3)) /
        2                 2     2
    (-x1  + 2 x1 x2 - x2  - y1  +
                 2
      2 y1 y2 - y2 )}}

FortranForm[%]>> c:\\temp4.for
```

(b)

定をメニュー「表示(V) マウスポインタの座標を表示(C)」から座標を表示するにセットします. 縞の間隔をマウスの座標から読み取ればよいのですが, 実際やって見ると縞と縞の間隔の垂線が定められず, なかなか正確に読み取れません.

そこで, **図 5** のように一つの縞上の 2 点 A, B ともう一つの縞の上の C 点の 3 点の座標を読み取り, **図 6** のように点 C を通り直線 AB に垂直な直線と直線 AB との交点 D をもとめます. CD の長さが求める縞の間隔の 2 本分に相当します. この縞の上の 3 点の座標はマウスを動かしてピクセル単位で正確に求めることができます.

点 A, B, C の座標を入力したら先の距離が求まる式を得るには, 高等学校で習ったことを思い出していただければいいのです. しかし, 具体的な式を求めることはなかなか面倒です. 2 元連立方程式を何回か解かなければなりません.

そこで, Mathematica を使いました. その様子を**図 7** に示します.

ここで, A 点の座標を (x 1, y 1), B 点を (x 2, y 2), C 点を (x 3, y 3) としました. Mathematica で得られた式を Fortran または C の形でセーブしてプログラム作成に利用します. Mathematica 上で,

**FortranForm[%]>> c:¥¥temp.for**

または,

**CForm[%] >> c:¥¥temp.c**

とすると, **図 8** に示すファイルが得られます.

筆者の場合は, Fortran でプログラムを作りました. プログラムを**図 9** に示します.

List と Rule を削るだけでプログラムの完成です.

これで, **図 5** の顕微鏡写真より必要なデータが得られます. しかし, ここで, 得られる値は相対的なもので, 絶対値は大きさのわかったものを同じ条件で顕微鏡撮影をし, 同様の方法で計測したものからの比により決めなければなりません.

**図10** に異なる解像度で 1 mm のスケール(ものさし)を撮影した写真を示します. この各画像を先の方法で相対距離を求めたところ, 解像度とスケール間のピ

〈図 8〉Mathematica から出力された式

Fortran 形式の場合

```
List(List(Rule(x,-((y1 - y2)*(x2*y1 - x1*y2)/
   -        (-(x1**2) + 2*x1*x2 - x2**2 - y1**2 + 2*y1*y2 - y2**2)) -
   -     (-x1 + x2)*(-(x1*x3) + x2*x3 - y1*y3 + y2*y3)/
   -      (-(x1**2) + 2*x1*x2 - x2**2 - y1**2 + 2*y1*y2 - y2**2)),
   -   Rule(y,-((-x1 + x2)*(x2*y1 - x1*y2)/
   -        (-(x1**2) + 2*x1*x2 - x2**2 - y1**2 + 2*y1*y2 - y2**2)) -
   -     (-y1 + y2)*(-(x1*x3) + x2*x3 - y1*y3 + y2*y3)/
   -      (-(x1**2) + 2*x1*x2 - x2**2 - y1**2 + 2*y1*y2 - y2**2))))
```

C 形式の場合

```
List(List(Rule(x,-((y1 - y2)*(x2*y1 - x1*y2)/
      (-Power(x1,2) + 2*x1*x2 - Power(x2,2) - Power(y1,2) + 2*y1*y2 -
        Power(y2,2))) - (-x1 + x2)*(-(x1*x3) + x2*x3 - y1*y3 + y2*y3)/
    (-Power(x1,2) + 2*x1*x2 - Power(x2,2) - Power(y1,2) + 2*y1*y2 -
      Power(y2,2))),Rule(y,-((-x1 + x2)*(x2*y1 - x1*y2)/
      (-Power(x1,2) + 2*x1*x2 - Power(x2,2) - Power(y1,2) + 2*y1*y2 -
        Power(y2,2))) - (-y1 + y2)*(-(x1*x3) + x2*x3 - y1*y3 + y2*y3)/
    (-Power(x1,2) + 2*x1*x2 - Power(x2,2) - Power(y1,2) + 2*y1*y2 -
      Power(y2,2)))))
```

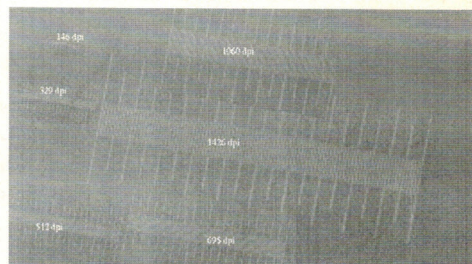
〈図 9〉Fortran によるプログラム

```
      REAL*8 X1,X2,X3,X4,Y1,Y2,Y3,Y4,XYLENG
      READ(*,*) X1,Y1,X2,Y2,X3,Y3
      X4=-((Y1 - Y2)*(X2*Y1 - X1*Y2)/
     -        (-(X1**2) + 2*X1*X2 - X2**2 - Y1**2 + 2*Y1*Y2 - Y2**2))
     -     (-X1 + X2)*(-(X1*X3) + X2*X3 - Y1*Y3 + Y2*Y3)/
     -      (-(X1**2) + 2*X1*X2 - X2**2 - Y1**2 + 2*Y1*Y2 - Y2**2)

      Y4=-((-X1 + X2)*(X2*Y1 - X1*Y2)/
     -        (-(X1**2) + 2*X1*X2 - X2**2 - Y1**2 + 2*Y1*Y2 - Y2**2))
     -     (-Y1 + Y2)*(-(X1*X3) + X2*X3 - Y1*Y3 + Y2*Y3)/
     -      (-(X1**2) + 2*X1*X2 - X2**2 - Y1**2 + 2*Y1*Y2 - Y2**2)

       XYLENG=SQRT( (X4-X3)*(X4-X3) + (Y4-Y3)*(Y4-Y3) )

       WRITE(*,*) XYLENG
       STOP
       END
```

(a)

(b)

〈図11〉◀
Mathematicaによる画像解析の例(フーリエ変換)

〈図10〉 各解像度によるスケール

クセル値は見事に比例関係をもっていました.

当たり前のことですが，責任を持った計測をするにはこのようなこともきちんと確かめておかなければなりません．もちろん，各画像取り込みの際に解像度を記録しておく必要があります．

フィルムスキャナで画像を取り込む場合，写真がきれいに撮れていれば，任意の部分を高解像度で取り込めます．そこで，先の顕微鏡写真の縞模様(周期構造)がはっきり見えている部分を取り込んで，数値に変換し，Mathematicaで2次元フーリエ変換した例を**図11**に示します．

縞模様がみごとにパワースペクトルに変換されています．この場合，パワースペクトルは，ビデオキャプチャ・ボードの項で述べた液晶の光散乱に相当します．ビデオキャプチャ・ボードの項では実験結果を示したのですが，この場合は計算により算出したものです．

この周期構造(縞の間隔)が分子程度の大きさになった場合は，パワースペクトルはX線回折の像に対応します．

## ● 最後に

ディジタルカメラは，そもそも始めからディジタル化されていますから，コンピュータに取り込むことも比較的簡単でしょう．しかし，普及していないこともあり，値段が高いです．普及とともに入手しやすい価格になる兆しは見えてきていますので，今後が楽しみです．

ビデオキャプチャ・ボードは画像取り込みに要する作業時間が短く，価格もお手頃です．しかし，取り込んだ画像の質はほかの方法と比べるとやや劣ります．これも，今後の技術開発により大きく改善されるのではないでしょうか．

PhotoCDは，CD-ROMドライブがあれば，現在のCD-ROMドライブはほとんどPhotoCDに対応していますから，それ以上の投資が必要ありません．現在，CD-ROMドライブが付いている環境は普通になりつつあります．

コスト的には大変優れています．また，画像の質も抜群に優れています．しかも，画像の保存という立場から考えるとCD-ROMに保存できるわけですから大変優れています．しかし，PhotoCDを作るには外注しなければなりませんし，できあがりまでに時間がかかります．

フィルムスキャナは，スキャナを手にいれる費用がかかります．しかし，実売価格は20万円を切っています．そして，画像の解像度は自由に調整が可能です．自分で現像が可能な白黒写真ならば写真撮影からディジタル画像を得るまでに1～2時間ぐらいでできます．外に出すことが難しい企業秘密がかかわるような画像や急ぐ必要がある画像の処理には適していると思います．

ディジタルカメラについては今回は試用できませんでしたが，今後機会があればぜひ試してみたいと思っています．PhotoCDはハード的なことがほとんどありません．CD-ROMになった画像をいかに活用するかにしぼられます．したがって，画像を解析する方法などについての入門的な話は，ほかの章やその他の参考書を参考にしてください．

◆ 参考文献 ◆

(1) 臼田・井上・伊藤；パソコンによる数式処理ソフト活用法，1994年，CQ出版(株)．
(2) 早川美徳；Mathematicaによる数値計算の実際と計測・制御への応用，インターフェース，1995年5月号，CQ出版(株)．

# Appendix HP VEEによる計測システム開発の効率化

柏木伸之

コンピュータを用いた自動計測の分野において，これまではテキスト形式のプログラミング言語によるシステムの開発が主流でした．しかし，プログラミング言語を使用した場合，プログラムのコーディング，デバッグにかなりの工数が必要とされていました．ここで紹介するHP VEE(Visual Engineering Environment)は，グラフィカルプログラミング言語を採用しており，計測システム開発期間を大幅に短縮します．

また，Windows版ではDDE(Dynamic Data Exchange)をサポートしているので，計測したデータをそのままWindowsのアプリケーションに渡し，アプリケーション側でデータの解析，グラフ化するなども可能です．

〈図1〉フィルタの特性を調べるシステムの例

## ● グラフィカルプログラミング言語

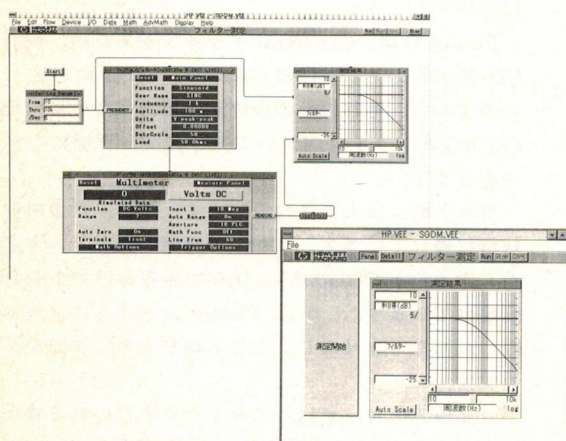
グラフィカルプログラミング言語のコンセプトは，ノンプログラミングの環境で自動計測システム開発を可能にするというものです．

テキスト形式のプログラミング言語を使用して開発を行う場合，まず，システムの構成，目的に合わせたシステムのブロックダイヤグラムを書き，それに基づきプログラムをコーディングしていきます．HP VEEでは，そのブロックダイヤグラムを書く要領で，HP VEEにあらかじめ用意されているオブジェクトと呼ばれるアイコンをHP VEEの作業エリアに配置し，線で結ぶだけでプログラムは完成します．

オブジェクトには，フロー制御，計測器制御，計算/解析，表示などの機能をもったものがあります(**表1**参照)．

## ● プログラム開発例

HP VEEを使った簡単なプログラム開発の事例を紹介します．

ここでは，ファンクションジェネレータの周波数をスイープさせ，ディジタルボルトメータでフィルタの周波数特性を調べるというプログラムを作ります．**図1**のように五つのオブジェクトを選択，配置し，結線

〈表1〉解析用オブジェクト

| Formula | xor | fracPart(x) | Power: | cot(x) |
|---|---|---|---|---|
| Elementary Math: | not | Complex Parts: | sq(x) | asin(x) |
| + | Bitwise: | j(x) | sqrt(x) | acos(x) |
| − | bit(x, n) | re(x) | cubert(x) | atan(x) |
| * | bits(str) | im(x) | recip(x) | acot(x) |
| / | setBit(x, n) | mag(x) | log(x) | atan2(y, x) |
| △ | clearBit(x, n) | phase(x) | log 10(x) | Hper Trig: |
| mod | bitAnd(x, y) | conj(x) | exp(x) | sinh(x) |
| div | bitOr(x, y) | String: | exp 10(x) | cosh(x) |
| Relational: | bitCmp(x) | strUp(str) | Polynominal: | tanh(x) |
| == | bitShift(x, y) | strDown(str) | 1:poly(x,[a0 a1]) | coth(x) |
| != | Real Parts: | strRev(str) | 2:poly(x,[a0 a1 a2]) | asinh(x) |
| < | abs(x) | strTrim(str) | 3:poly(x,[a0 a1 a2 a3]) | acosh(x) |
| > | signof(x) | strLen(str) | N:poly(x, [a0 a1 ... aN]) | acoth(x) |
| <= | ordinal(x) | strFrom | Trig: | Time & Date: |
| >= | round(x) | Thru(str, from, thru) | sin(x) | now() |
| Logical: | floor(x) | strFromLen(str, from, len) | cos(x) | wday(data) |
| and | ceil(x) | strPosChar(str, char) | tan(x) | mday(date) |
| or | intPart(x) | strPosStr(str1, str2) | | |

するだけでプログラムは完成します．

これは，フロー制御オブジェクトでスイープ周波数を設定，計測器パネルで各計測器を制御，表示のオブジェクトに測定結果を表示させます．このように計測器の制御は，400を超える計測器パネルを標準で用意されていますが，計測器パネルのない計測器でも“Direct I/O”オブジェクトを使用すれば，HP-IB，RS-232C，VXIなどのどんな計測器も制御が可能です．

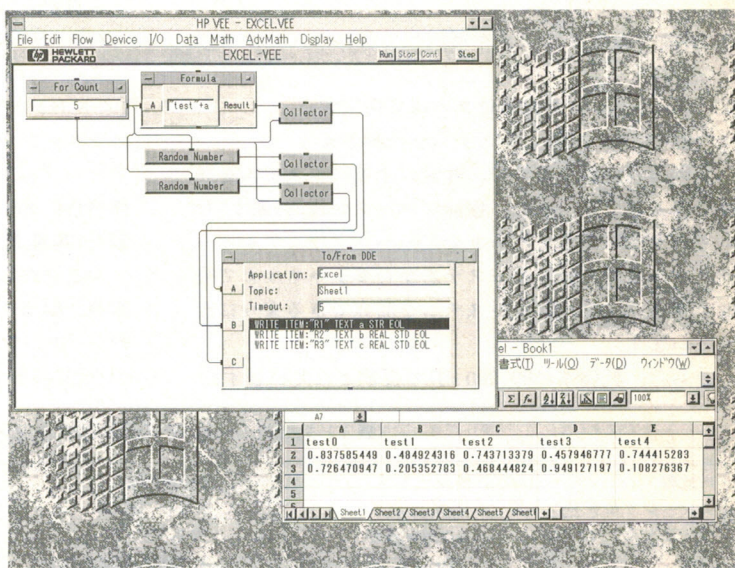
また，“詳細表示”，“パネル表示”と二つの表示を設定できます．“パネル表示”には，オペレータに対して必要最小限のオブジェクトのみを表示させることができます．これで，簡単にユーザーインターフェースも作成でき，オペレータが不用意にプログラムを変更してしまうこともありません．

このようなプログラムも，一般的なプログラム言語を使用して開発するのと比較すると，簡単に完成します．

### ● ほかのアプリケーションとのデータの通信

HP VEE for Windowsの場合，ほかのWindowsアプリケーションソフトウェアに，直接計測データを送ることができます．

**図2**は，データをMS-Excelに送っている例です．このように，ほかのアプリケーションにアクセスして，それらの特殊機能，すなわちデータベース，レポートジェネレータおよびワードプロセッサを活用することができます．このように，ほかのアプリケーションへのアクセス機能は，DDE(Dynamic Data Exchange)として知られています．

**〈図2〉**
**エクセルにデータを送っている様子**

**問い合わせ先：**
日本ヒューレット・パッカード㈱
☎ 0120-421-345
FAX 0120-421-678

month(date)
year(date)
dmyToDate(d, m, y)
hmsToSec(h, m, s)
hmsToHour(h, m, s)

**Array:**
init(x, value)
rotate(x, numElem)
concat(x, y)
sum(x)
prod(x)
sort(x)

**Matrix:**
det(x)
inverse(x)
transpose(x)
identify(x)
minor(x, row, col)
cofactor(x, row, col)
matMultiply(x, y)
matDivide(x, y)

**Calculus:**
integral(x)
deriv(x, 1)
deriv(x, 2)
deriv(x, order)
defintegral(x, a, b)
derivAt(x, 1, pt)
derivAt(x, 2, pt)
derivAt(x, order, pt)

**Regression:**
linear
logarithmic
exponential
power curve
polynomial

**Data Filtering:**
polySmooth(x)
meanSmooth(x, numPts)
movingAvg(x, numPts)
clipUpper(x, a)
clipLower(x, a)
minIndex(x)
maxIndex(x)
minX(x)
maxX(x)

**Probability:**
random(low, high)
randomize(x)
randomSeed(seed)
perm(n, r)
comb(n, r)
gamma(x)
beta(x, y)
factorial(n)
binomial(a, b)
erfc(x)
erf(x)

**Statistics:**
min(x)
max(x)
median(x)
mode(x)
mean(x)
sdev(x)
vari(x)
rms(x)

**Freq. Distribution:**
linMagDist(x, from, thru, linStep)
logMagDist(x, from, thru, logStep)

**Bessel:**
j0(x)
j1(x)
jn(x, n)
y0(x)
y1(x)
yn(x, n)
Ai(x)
Bi(x)

**Hyper Bessel:**
i0(x)
i1(x)
k0(x)
k1(x)

**Signal Processing:**
fft(x)
ifft(x)
convol(a, b)
xcorrelate(a, b)
bartfet(x)
hamming(x)
hanning(x)
blackman(x)
rect(x)

# Appendix Visual Designer：減衰率測定装置への応用

臼井重徳

現代は，あらゆる面において経費削減・工数短縮が絶対命題になってきています．しかも，性能およびパフォーマンスは従来以上の能力を求められています．ここでは，そのような場合に最適な，計測・解析・制御・表示用の「グラフィックアプリケーション・システム開発ツールソフトウェア」Visual Designer を紹介します．

このソフトウェアは従来のプログラミングが不要で，3 ステップの簡単操作でシステムが完成してしまいます．

(1) 必要な機能ブロックを選択して各パラメータを設定する
(2) データフローを矢印付きの線で指定する
(3) 起動ボタンを押しシステムを始動させる

このように簡単なので，非常に迅速にシステムを開発できるのです．また，機能ブロックが豊富(**表 1**)なので，サポートするボードなどのハードウェアがポータブルから超高速，小システムから多チャネルまで充実しており，高機能なシステムを開発できるようになっています．

さらに，低価格(189,000 円)で提供されています．

## ● 応用例

さて，この Visual Designer を用いた応用例として，減衰信号波形の「減衰率測定装置」を紹介します．0 Hz 以上の，海の波，地震波，音声波形，機械振動，レーダー波，超音波，画像信号など応用も多種に渡って考えられ，信号波形収集部分を変えればどの応用においても方法は同一といえます．信号波形に含まれる最大周波数に応じて，その電気信号の計測用装置またはPC の計測用ボードを選定します．

これから説明する応用例は，ある容器の中で回転する特殊な球の発生する音の減衰率測定装置です．

ここでは，最大 A-D 変換スピードが 45 kHz から 100 MHz まで用意されているインテリジェント インストルメンテーション社の各種ボードの中から，16ch アナログ入力，2ch アナログ出力その他を搭載した低価格(99,000 円)の PCI-20428W-1 カードを使用しました(**写真 1**，**表 2** 参照)．

マイクからの音声帯域信号なので，サンプリング周波数：44.1 kHz で 5 秒間のデータ収集を行います．そのシーケンスを次に示します．

(1) 信号波形の捕捉

〈表 1〉 Visual Designer の広範囲な機能ブロック

**▶比較**
スレッショルドアラーム
ウィンドウアラーム
比較

**▶データ変換**
積算
連結
カット
デシメーション
ディレイ
単位変更
マルチプレクス
パック
アンパック
形式変換

**▶ DDE 機能**
DDE クライアント
DDE サーバー

**▶表示**
バーグラフ
マルチチャネル・プロット
プロット
表示器
アナログメータ
パネル
チャート
X/Y プロット

**▶ DSP 機能**
パワースペクトラム
ディジタルフィルタ
FFT

**▶ファイル入出力**
ファイル名コントロール
バイナリファイルの書き込み
ASCII ファイルの書き込み
バイナリファイルの読み出し
ASCII ファイルの読み出し

**▶ GPIB 通信**
IEEE-488

**▶入出力**
アナログ入力
アナログ出力
ディジタル入力
ディジタル入力ポート
ディジタル入力 ZPB6064
ディジタル出力
ディジタル出力ポート
カウンタ
8254 カウンタ
周期の測定
レート発生器
熱電対入力

**▶ DMA 転送入出力**
アナログ入力 DMA
アナログ出力 DMA
ディジタル入力 DMA
ディジタル出力 DMA
カウンタ DMA

**▶論理演算**
AND
NAND
NOR
NOT
ワンショット
OR
SR フリップフロップ
XOR

**▶数値演算**
絶対値
加算
クリッピング
微分
除算
積分
自乗 XY
平方根
乗算
逆数(1/X)
Log 10
Ln
移動平均
10^x
X^Y
mX＋b
平方
減算

**▶測定**
ヒストグラム
最大値
最小値
ピーク値
平均/総和
RMS

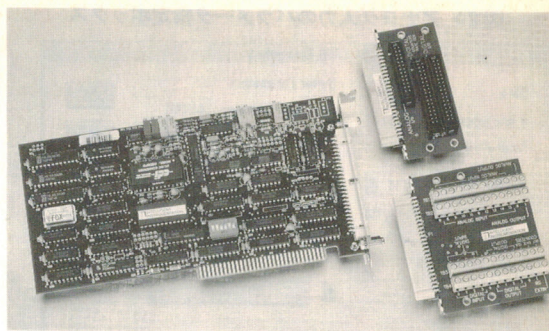
**▶その他の機能**
カット
エクスポート
抜き出し
フィードバック
インポート
ラッチ
ロード
PID
リターン
RUN コントローラ
タイム
タイマ
待ち

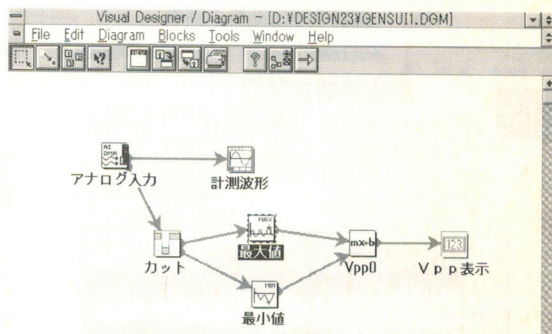
**▶シリアル通信**
シリアル

**▶ユーザインターフェース**
アルファニューメリック入力
オーディオ告知器
メッセージビープ
数値編集
スライダ
スイッチバー
ユーザープロンプト

**▶波形発生器**
常数
パルス
ランダム数
サイン波

〈表2〉PCI-20428W-1の仕様

| パラメータ | 条　件 | 仕　様 |
|---|---|---|
| アナログ入力 | | |
| ▶チャネル数，分解能 | シングルエンド/差動入力 | 16/8，12ビット(4096の一つ) |
| ▶A-D電圧レンジ | | 0～5V，0～10V，+/−5V，+/−10V |
| ▶過電圧保護 | パワーオン時，パワーオフ時 | +/−35V，+/−20V |
| ▶コモンモード・リジェクション | 60Hz，100Ω不平衡，ゲイン100 | −92dB(0.05LSB/V) |
| ▶直線性エラー | | +/−0.024%FS |
| ▶スループット(マルチチャネルで) | DMA，トリガー付DMA使用時 | 100kHz |
| アナログ出力(-1，-2のみ) | | |
| ▶チャネル数，分解能 | | 2，12ビット(4096の一つ) |
| ▶電圧レンジ | | 0～10V，+/−5V，+/−10V |
| ▶スループット | DMA使用時 | 100kHz |
| ディジタル入出力 | | |
| チャネル数 | | 8入力，8出力 |
| カウンタ | | |
| チャネル，クロックスピード | | 1(16ビット)，8MHz |
| 電源条件 | +5V | 1A |

〈写真1〉PCI-20428Wシリーズの概観

〈図1〉最大波高値Vppの算出までのブロック図

(2) 捕捉した信号波形をグラフ表示する

(3) 5秒間を50等分し，B0からB49のブロックに分割する．

(4) B0，B10，B20，…B40の5ブロックの最大波高値Vppを抽出する．

ブロック$n$の最大波高値$Vpp_n = MAX_n - MIN_n$

(5) 単純な信号減衰率DR(Damped Ratio)を計算する．

$DR = [Vpp_0 - Vpp_4]/10$

(6) 平均信号減衰率ADRを計算する．

$DR_{n+1} = Vpp_{n+1}/Vpp_n$

$ADR = \sum_{n=0}^{3} DR_{n+1}/4$

(7) 計算した単純な信号減衰率と，平均信号減衰率を信号波形の上に表示する．

**図1**は，最大波高値$Vpp_n$の算出結果を表示するまでを示しています．計測された信号波形データは，まず表示されます．そして，カット機能により最初のB0ブロックの0.1秒間のデータが切り出され，そのデータ列の最大値と最小値が測定され，波高値＝最大値−最小値を最大波高値$Vpp_0$として算出します．

各ブロックのパラメータを設定する例として，PCI-20428W-1のアナログ入力部のダイアログボックスを

## コラム●ドイツMESSCOMPショー

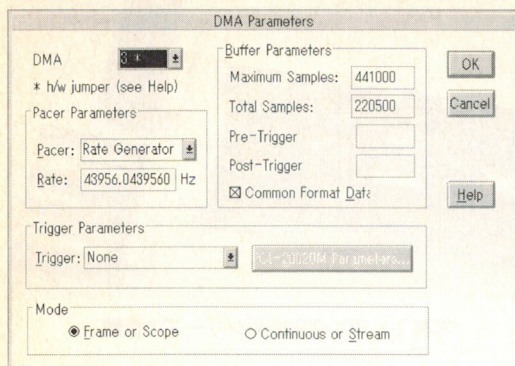
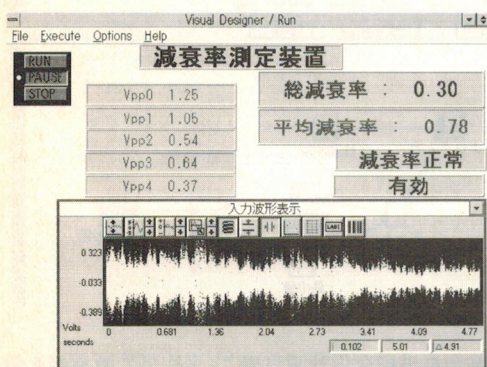
このドイツMESSCOMPショーは1994年秋に開催され，世界の有名な計測ソフトを提供している8社のメーカーの技術者が参集し，「コイン弁別システム」を3時間以内に作成するという命題に挑戦しました．この中で，インテリジェント インストルメンテーション社のVisual Designerが1時間45分で完成し，従来からの計測ソフトメーカーを抑えて優勝しました．

そこでは，数種類のコインを大理石のテーブル上に落とし，その際，(1) 音をマイクロホンで電気信号に変換し，(2) A-D変換し，(3) FFT変換し，(4) 音の振動スペクトラムを解析して，いわば，パターン認識用のテーブルを各コインに対して用意します．

このコインパターン認識の中核機能のほかに，アンプでの増幅，パターン認識のロジック，認識結果の表示，さらに進んで，音声による認識結果の報知などの機能を付加したシステムが作成されたのです．

従来かかっていた数ヶ月という工数が1000分の1と大幅に短縮されています(詳細は文献を参照)．

〈図 2〉 アナログ入力のパラメータ指定ボックス

〈図 4〉 減衰率測定結果の画面

〈図 3 (a)〉 ブロック図の最初部分

〈図 3 (b)〉 ブロック図の最終部分

**問い合わせ先：**

インテリジェント インストルメンテーション
〒222 横浜市港北区新横浜 1-3-10
☎045-474-4162　Fax：045-474-4167

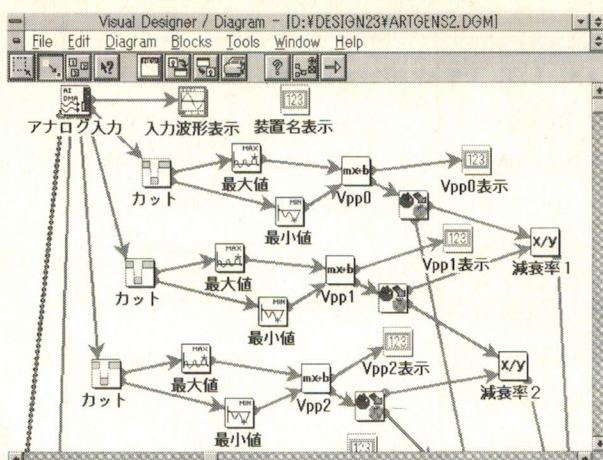
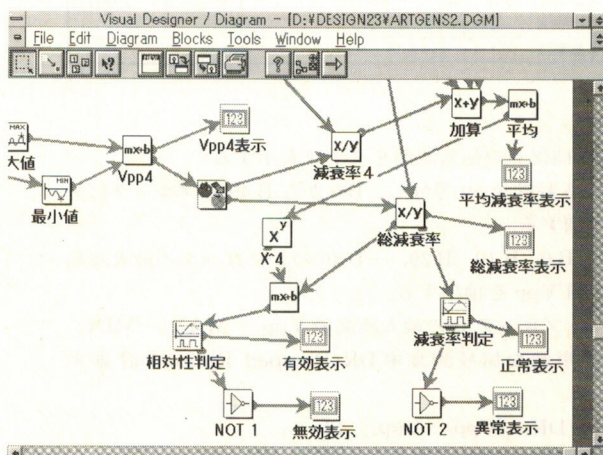



図 2 に示します．

このブロック図に必要なシーケンスを追加して作成したのがブロック図 3 (a)とブロック図 3 (b)です．全体のブロック図の最初の左上の部分が図 3 (a)で，最後の右下の部分が図 3 (b)です．

このようにして作成したブロック図がシステムの設計図であり，RUN 環境上で起動すればシステムが動作します．表示部分はシステムが動作後にマウスを使用し，微調整すればシステムが完成します．

完成し，減衰率測定結果を表示している画面を図 4 に示します．

このようにユーザ自身のシステムの詳細ブロックが概念としてできあがっていれば，Visual Designer を使用して，非常に簡単にシステムを構築，微調整，完全動作まで行い，開発を極めて短時間に完成させることができます．

日本語ユーザーズマニュアルも完備しています．

◆ 参考文献 ◆

(1) 計測・解析ソフトウェア〈VisualDesigner〉MESSCOMP の計測競技で優勝，ELECTORNICS UPDATE 1995-3.

新連載

# Delphiプログラミング

## ① 初めてのプログラミングとTPWからの移行

岡本 安晴

Borland社が用意したクラスライブラリOWLとともにTurbo Pascal for Windowsが発売されたときは，Windows時代になってもTurbo Pascalは使える言語だと思われていました．

ところが，Borland Pascalの日本語版の発売が見送られ，結果的にTurbo Pascal for Windowsのまま打ち止めになるかのように思える状況が続いていました．

しかし，今年の春に米国で発売されたDelphi(日本語版の発売は秋頃)の登場により，この状況が一変しました．

Delphiは，言うなればObject Pascalを拡張したOOP言語を用いたデータベース・アクセス機能をもつビジュアルプログラミング・ツールで，機能的にはVisual BasicのすべてとVisual C++のほとんどを含んだものと言えます．なお，Client/Server版ではサーバーのアプリケーションを開発することもできます．

Delphiでは，Delphi Componentと呼ばれるカスタムコントロールの使用/新規作成をサポートし，またこれらは，Delphi Component Library(DCL)というライブラリに含まれ，MicrosoftのVBXをサポートします．

言語としてはObject Pascalであるため，Delphi Componentを継承することができ，Visual Basicのオブジェクトを基本としたプログラミングに比べて，高度で効率的なオブジェクト指向プログラミングを実践することもできます．

さらに，データベース・アクセスツール(こちらが本命？)としてみれば，データベースにアクセスするためのコントロールが標準でサポートされており，さらに英語版Delphiには，Local InterBase(スタンドアロンのSQLデータベース)も付属しています．

この連載では，Delphiのプログラミング入門からデータベースのアクセスツールの制作までを取り上げます．　**(編集部)**

---

DelphiはVB(Visual Basic)より高機能であり，VC(Visual C++)よりやさしいと評価されているデータベースアクセス機能を含んだObject Pascalのビジュアルプログラミング・ツールです(1)．

この連載ではDelphiを用いていろいろなプログラミング例を紹介していきます．

第1回目はDelphiによるビジュアルなプログラミングの概要を知るために簡単なプログラムを紹介します．

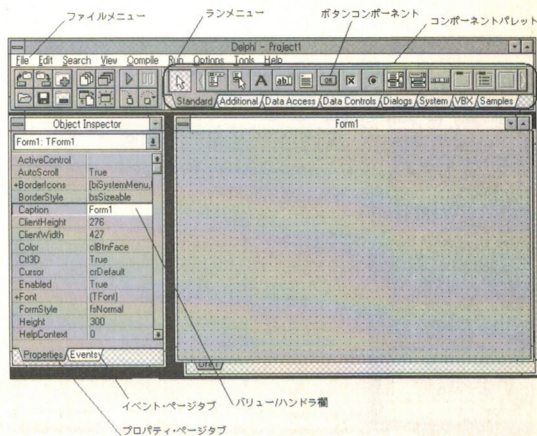
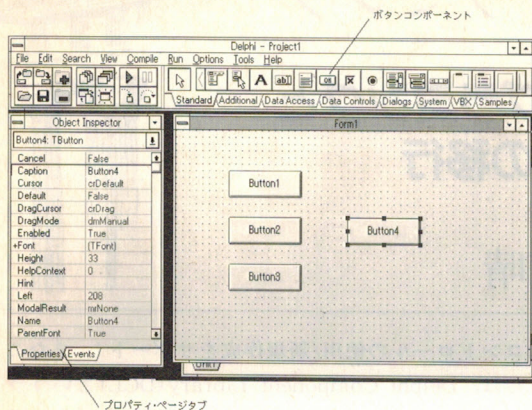
また，TPW(Turbo Pascal for Windows)ユーザーの方々のために，TPWで書かれたプログラムは，必要ならば，簡単な変更を加えるだけでDelphi上で走らせることができるという例を，その方法を交えながら説明します．

## Delphiの簡単な使用例

### ● Delphiの操作環境

Delphiのアイコンをダブルクリックすると，**図1-1**のような画面になります．

まず，メニューバーの[File]をクリックして現れるメニューから［New Project］を選んでクリックします．このような操作は，以後［File|New Project］をクリックというように記述します．

〈図1-1〉Delphiの環境

〈図1-2〉 ボタンコンポーネントをフォームに貼りつける

Delphiでは，中心となるプログラムとそれに結び付けられているファイル(ユニットファイル)をまとめてプロジェクトと呼びます．

中心となるファイルはプロジェクトファイルと呼ばれ，そのファイル名はプロジェクト名に拡張子**.DPR**の付けられたものとなります．

ユニットファイルの拡張子は**.PAS**です．Delphiでは，拡張子**.DPR**の付けられたファイルはプロジェクト作成の過程においてDelphiによって自動的に変更が加えられるもので，プログラミングはユニットファイルに対して行うことになっています．

しかし，TPW(Turbo Pascal for Windows)など従来のPascalの場合と同じスタイルでプログラミングを行うときには，プロジェクトファイルに対してソースコードを書いていくこともできますが，これはDelphiでの標準的なやり方ではないとされています．

［New Project］をクリックすると新しいプロジェクトのひな型が用意されますが，これはプロジェクトファイルと一つのユニットファイルから構成されています．

このユニットファイルには，フォームと呼ばれるウィンドウが一つ対応しています．**Form1**のタイトルの付いたウィンドウがそれです．このウィンドウ(フォーム)を使ってビジュアルプログラミングを行います．

プログラミングを始める前に，この新しく用意されたプロジェクトを適当なディレクトリに移します．

一つのディレクトリに一つのプロジェクトをまとめて置くようにすると，ファイル名の管理がしやすくなります．［File|Save Project As］をクリックすると，まずユニットファイルの名前を指定するように求められます．

移したいディレクトリを指定して［OK］ボタンをクリックします．次に，プロジェクトの名前を指定するように求められますので，ユニットと同じディレクトリを指定して［OK］ボタンをクリックします．

### ● ボタンコンポーネントを使ったビジュアルプログラミング

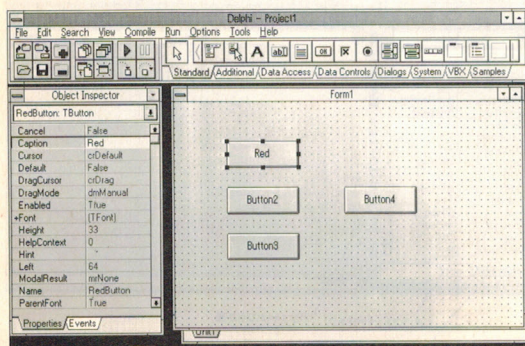
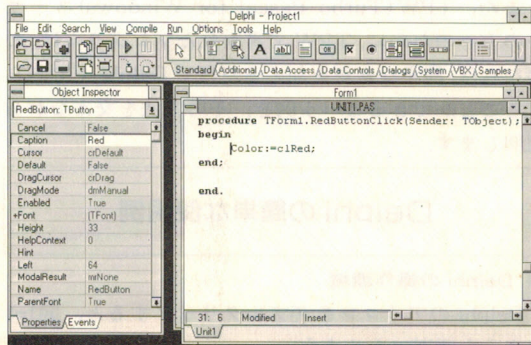
コンポーネントパレット上のボタンコンポーネント(**図1-1**)をクリックしてからフォームウィンドウ上の左上の方をクリックすると，フォームのその場所にボタンがコピーされます(**図1-2**)．

ボタンなどのコンポーネントパレット上の部品は，コンポーネントと呼ばれています．コピーされたボタンはマウスでドラッグすることにより，位置や大きさを変えることができます．

コンポーネントパレット上のボタンコンポーネントをクリックしてからフォーム上の適当な場所をクリックして，ボタンコンポーネントのコピーをとるという操作をさらに3回繰り返して，**図1-2**のように四つのボタンを置きます．

次に，フォーム上の［Button1］をクリックしてアクティブ(枠で囲まれることによって示される)にしてから，フォーム左側にあるオブジェクトインスペクタのプロパティページ上の**Name**欄をクリックし，**Name**欄の右側のバリュー欄にキーボードからの入力ができるようにします．

プロパティページがイベントページの後ろに隠れて

〈図1-3〉 Captionを入力

〈図1-4〉 コードを入力

いるときは，プロパティページ・タブをクリックするとプロパティページが表示されます．

**Name** 欄に **RedButton** と入力すると，［Button1］を記述しているオブジェクト（コンポーネントのコピーをとると自動的にユニットファイルに宣言される）の名前が **RedButton** となります．

続いて，**Caption** 欄をクリックしてそのバリュー欄に **Red** と入力します．**Caption** に対して指定した文字列がボタン上に表示されます（**図1-3**）．

次に，イベントページ・タブをクリックしてイベントページを表示させます．［Red］ボタンをクリックしたときの処理を記述するため，**OnClick** に対するハンドラ欄をダブルクリックします．

**OnClick** は，マウスによるクリックというイベントのことです．エディタによって **UNIT1.PAS** ファイルが表示されますので，**図1-4** のように **begin** と **end** の間に，

**Color :＝clRed ;**

と入力します．

Pascalでは，

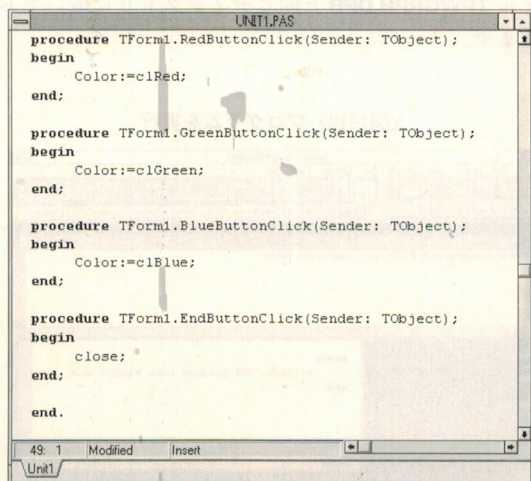
**:＝**

は代入を表します．ここで，**clRed** は赤色を表す値です．したがって，**Color** に **clRed** が代入されるとフォームの色が赤色になります．

次に，エディタの後ろに隠れているフォームのタイトルバーをクリックしてフォームウィンドウを前に出し，［Button2］と［Button3］に対しても同じような操作を繰り返します．

**Name** をそれぞれ **GreenButton**，**BlueButton** と指定し，**OnClick** イベントに対しては

**Color :＝clGreen ;**

あるいは

**Color :＝clBlue ;**

と記述します．**clGreen**，**clBlue** は緑色，青色を表す値です．

［Button4］に対しては，**Name** は **EndButton**，**Caption** は **End** として，このボタンのクリックによってプログラムが終了することを表示します．クリックに対して終了するために **EndButton** の **OnClick** イベントに対する記述は，

**Close ;**

となります．

**Close** の実行によって，フォームは閉じられてプログラムは終了します．

以上の操作によって，各ボタンのクリックに対する処理は**図1-5** のように記述されることになります．

**UNIT1.PAS** を表示しているエディタの大きさは，マウスのドラッグによって変えることができます．

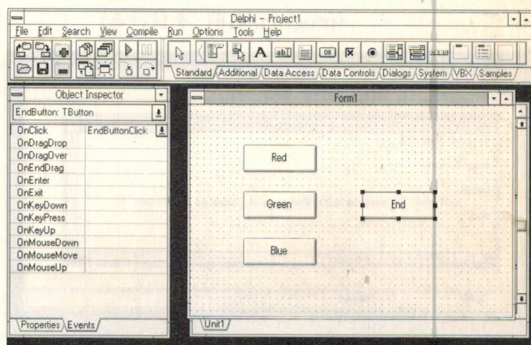
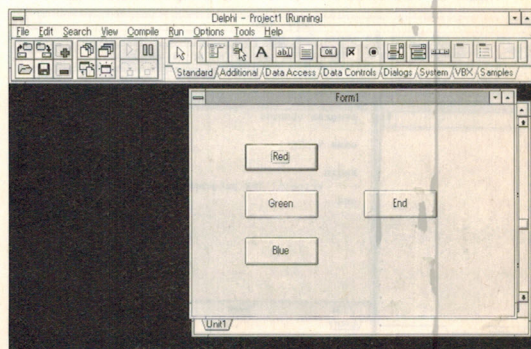
フォームのウィンドウをクリックするとフォームが前に現れますが，**図1-6** のようになっていると思います．

このフォームはビジュアルプログラミングのときに，そのときのプログラミングの内容に対応して Delphi によって表示されているもので，実行時のものとは別のものです．

［f･9］キーを押すとプロジェクトが実行されます．実

〈図1-5〉 各ボタンの処理を記述した

```
procedure TForm1.RedButtonClick(Sender: TObject);
begin
    Color:=clRed;
end;

procedure TForm1.GreenButtonClick(Sender: TObject);
begin
    Color:=clGreen;
end;

procedure TForm1.BlueButtonClick(Sender: TObject);
begin
    Color:=clBlue;
end;

procedure TForm1.EndButtonClick(Sender: TObject);
begin
    close;
end;

end.
```

〈図1-6〉 完成したフォーム

〈図1-7〉 プロジェクトを実行

行時の画面は**図1-7** のようになります．

オブジェクトインスペクタが消えています．表示されているフォームは，プロジェクトの実行によって生成されているウィンドウです．

フォーム上の三つのボタンをそれぞれクリックすると，ボタンに対応してフォームの色が変わります．

［End］ボタンを押せば，プログラムは終了します．Windows のプログラムですので［GRPH(ALT)］＋［f･4］キー操作でも終了します．

## TPW プログラムの Delphi 用への変更例

### ● Delphi でも WinCrt は使える

TPW で書かれたプログラムがユニットとして **WinCrt** だけを使用している場合は，そのままで Delphi 上でも実行できます．

ただし，TPW の場合は主プログラムのファイル拡張子は **.PAS** ですが，Delphi では **.DPR** となります．

また，TPW ではプログラム頭部は必要ありませんが，Delphi ではファイル名が**名前 .DPR** のとき同じ名前を用いて，

**program　名前；**

の形でプログラム頭部を置きます．

上のように変更したプログラム(**図1-8**)を，Delphi から［File|Open Project］によって読み込むと**図1-9** のようになります．［f･9］キーを押して実行すると**図1-10** のように CRT ウィンドウに **writeln** 手続きによる出力が表示されます．

### ● TPW のグラフィックスプログラムを Delphi 上で走らせるには

TPW で作成したグラフィックスプログラムを Delphi 上で走らせるときには，少し工夫が必要です．同じ名前のユニットでも，TPW の場合と Delphi の場合では内容が合わないものがあるようです．

この場合は，TPW の RTL(Run Time Library)のユニット・ソースコード内のユニット名を変更して使用すれば Delphi 上で走らせることができます．

**リスト1-1** のプログラムは，本誌 1995 年 8 月号の記事「Turbo Pascal for Windows を使った Windows プログラミング」に掲載されている TPW グラフィックスプログラムを Delphi 用に直したものの一部です．変更箇所は 3 行目からの **uses** 節だけです．

元のリストでは，

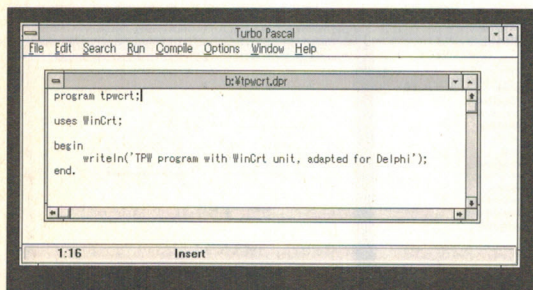
**uses WinCrt, WObjects, WinTypes, WinProcs, Strings ;**

となっているユニット名 **WinCrt** から **Strings** までを **WinCrt0** から **String0** までに変えてあります．

さらに，**WinCrt** から **Strings** までのユニットによって使用されていたユニット **WinDos** と **WinNls** も，名前を **WinDos0** と **WinNls0** とに変えて宣言しておきます．

それらの納められているソースコード・ファイル(RTL) **WINDOS.PAS** から **STRINGS.PAS** 内のユニット名もそれらにあわせて変更します．RTL はボーランド社から公開されています．

**uses** 節の変更部分においてユニットのソース・コードのファイルとして挙げられている **WINDOS.PAS** から **STRINGS.PAS** までのファイルはプロジェクトファイル **TPWGRPH.DPR** と同じディレクトリに置いておきます．

〈図1-8〉 TPW のプログラムの変更
(program 名前；を付ける)

〈図1-9〉 TPW のプログラムを Delphi に読み込む

〈図1-10〉 プログラムを実行

12 行目から後の部分は，TPW プログラムとまったく同じです．

[f·9] キーを押せば，このプログラム（プロジェクト）は Delphi 上で実行されます．

次回は，Delphi の機能によるグラフィクスプログラミングを紹介します．


◆ 参考文献 ◆

(1) 松原 敦；delphi，日経バイト，1995，3，136，234-247，日経 BP 社．

(2) Borland Delphi for windows: User's Guide，Borland International，Inc.，1995.


〈リスト1-1〉 グラフィックスプログラムはユニット名を変更（太字の部分）

```
 1  program  Tpwgrph;
 2
 3  uses     WinDos0 in 'WINDOS.PAS',
 4           WinNls0  in 'WINNLS.PAS',
 5           WinCrt0 in 'WINCRT.PAS',
 6           WinType0 in 'WINTYPES.PAS',
 7           WinProc0 in 'WINPROCS',
 8           Wobject0 in 'WOBJECTS.PAS',
 9           String0 in 'STRINGS.PAS';
10
11
12  const    plane_x_max = 350;
```

## コラム●日本語版 Delphi for Windows 早期体験版を使ってみる

先頃，Borland 社が始めた「日本語版 Delphi for Windows 早期体験プログラムキット」が入手できました（**図1-A**）．ここでは，簡単にこのキットに基づいての日本語版の紹介をしたいと思います．

### ● 英語版と日本語版の大きな違い

「早期体験プログラム」に含まれる **readme.txt** によれば，「日本語版 Delphi for Windows では，サポートツールのうち ReportSmith，Local InterBase Server，Database Desktop が添付されない予定」とのことです．

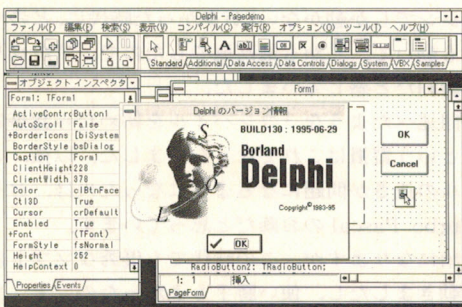
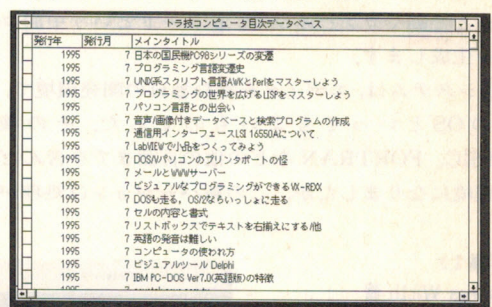
英語版には，Local InterBase Server が含まれているため，手軽に SQL に触れることができたのですが，残念なことです．

〈図1-A〉 Delphi 早期体験プログラム

英語版と「早期体験プログラム」の日本語版のグループを**図1-B**に示します．

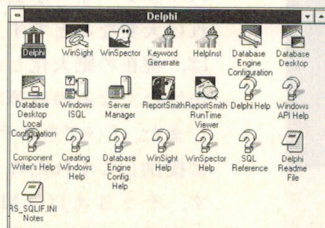
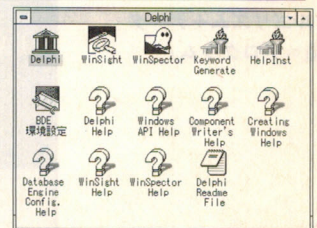
### ● 「早期体験プログラム」には，SQL Link 2.5 が含まれていない

Delphi で作成するアプリケーションは SQL Link 2.5 を使ってサーバーデータを操作することができますが，SQL Link 2.5 は早期体験プログラムには含まれないため，作成するアプリケーションが扱えるデータはローカルデータに限られました．

そこで，ローカルデータへのアクセスを試すために，トラ技コンピュータの目次データベース（Paradox 5.0 の DB ファイル）を表示するプログラムを制作してみました（**図1-C**）．

〈図1-C〉 DB ファイルの読み込み表示プログラム

〈図1-B〉
英語版と日本語版
（早期体験版）
のグループの違い

(a) 英語版

(b) 日本語版

連載

# パソコン草創期ソフトウェア回顧録

## 第3回　Turbo Pascalとの出会い

西田雅昭

### いろいろなPascal

連載の第1回目でも述べましたが，Pascalとの出会いが筆者のプログラムへの考え方を根本的に変えてくれました．すなわち，読みやすい，美しいプログラムを書くことに喜びを見い出したのです．

そして，アルゴリズムというものにも夢中になりました．これにはNiklaus Wirthの影響が大きく，長いこと金科玉条としていた，「Algorithms＋Data Structures ＝ Programs」は，今でも実際に役に立っています．

筆者が最初に使った本格的なPascalは，Pascal MT＋(MT micro systems，後にDigital Reseach)でした．これはちょっと大がかりなシステムですが，対話型のデバッグ環境のすごさに感激した記憶があります．

このPascal MT＋を使って実用的なプログラムを書いたこともありましたが，結構使い方の難しい処理系でした．そのため，後述するTurbo Pascal(Borland)が登場すると，筆者の頭の中からは消えてしまいました．

カリフォルニア大学サンディエゴ校が開発したUCSD-Pascalも，一つの時代を作ったコンパイラであったかもしれません．UCSD-Pascalはpシステムのもとで動くコンパイラで，pコードという中間コードを生成します．

pシステムは，エディタを含む独自の開発環境で，一種のOSといってもよいものでした．その後，BASIC，FORTRANなどの他の言語までを含んだ統一環境になりましたが，この新バージョンは処理があまりにも遅くて，筆者の禁煙をつぶすと同時に忘れてしまいました．

JRT-Pascalは安価なシステムで，ISAMもどきのファイルを作ることができる，かなり素晴らしい処理系でした．順当ならば，パソコン上のPascalの標準になったかもしれません．米国でもずいぶん評判になっていました．

筆者自身も，事務用のデータ処理プログラムを書くことが多かったので，ずいぶん勉強しました．しかし，衝撃的なTurbo Pascalの出現が，全てを変えてしまいました．

〈写真1〉
Niklaus Wirth著
片山卓也訳
「アルゴリズム＋データ構造＝プログラム」

### Turbo Pascalとの出会い

Turbo Pascalは，歴史に残るすばらしい処理系と言ってよいでしょう．エディタ，デバッガ，インタプリタならびにコンパイラの実行環境を統一して提供したのは，Turbo Pascalが初めてだったのではないかと思います．

しかも，コンパイル，リンクのスピードが考えられないほど速かったのです．またエディタも，当時のあらゆるエディタよりも優れたものでした．それでいて，360Kバイトのフロッピに入るというコンパクトなものでした．

マシンのファームウェアをいじり回すソフト(特定のマシンでしか動かない)が多かった時代に，どのマシンでも動いて，しかも最高速でしたので，アルゴリズムの勝利だと感じていました．

前回述べたような「コンパイルを始めたら食事に行く」という習慣はこれでなくなりました．以後，コンパイルの速さが問題になるようになったのは，おそらくTurbo Pascalのお蔭だと思っています．

余談になりますが，この時代でも優秀なソフトは軽快に動きましたし，使い勝手もよかったのです．当時のマシンパワーが低かったことを考え合わせると，これは大変なことです．この頃をよく知っている人ならば，現在の肥大化したアプリケーションに疑問を投げかけたくなるのは筆者だけではないでしょう．

また，「バグをエディタ上で指す」という環境を最初に提供してくれたのもTurbo Pascalだったと思います．そう言えば，Macintosh版のTurbo Pascalは，バグをしゃべってくれましたし，エディタ上で本当に「虫」が出てきたのにはびっくりしました．

パソコンの環境にあわせて，低レベルも処理できるようになっていましたので，アセンブラの必要な部分も，Pascalのコードの中で記述できました．Turbo Pascalのおかげで，アセンブラを使うのが下手になってしまいました．

米国では，Turbo Pascalで書いたPDSが数多く出現しましたし，Turbo Pascalのソースで提供するパッケージソフトもたくさんありました．そこで，dBASEのデータファイルをdBASE並みに扱えるソフト(ソース付き)を購入して，Xbaseファイルの内部構造を隅から隅まで知ることもできましたし，WordStarにそっくりなエディタ(ソース付き)を購入して，エディタの作り方も覚えました．

筆者は，この処理系に惚れ込んで，Windows時代が来るまでは小さなツールから大きなシステムまで，すべてTurbo Pascalで書いていました(ファイルのハンドリングの細やかさもスピードも抜群でした)し，雑誌などでも機会があれば，この処理系の素晴らしさを紹介しました．

多くの方が「西田にだまされて，Turbo Pascalを買ってしまった」と雑誌に書いておられたのを覚えています．

Turbo PascalにはCP/M用とMS-DOS用がありました．このバージョン3は大変よくできていたので，その後のバージョンアップで入手できなくなってずいぶんがっかりしました．

バージョン5は時代の流れに合わせて作られた製品でしたし，バージョン5.5からはOOP(オブジェクト指向)対応となり，Windows版も発売されました．しかし，この頃のバージョンアップにはちょっとついていけないように感じていました．

そして，筆者が初めてVisualBasic 1.0を手にしたときから，あれほど絶賛していたTurbo Pascalに別れを告げて，再びBASICの世界に浸り込むことになります．Windowsの世界では，Turbo Pascalをもってしても，とても勝負にならなかったのです．

最近，VisualBasicの対抗としてDelphi(Borland)が売り出されましたが，はたしてPascalへ戻ることがあるでしょうか．

〈写真2〉
Turbo Pascal version 3.0の英文マニュアル

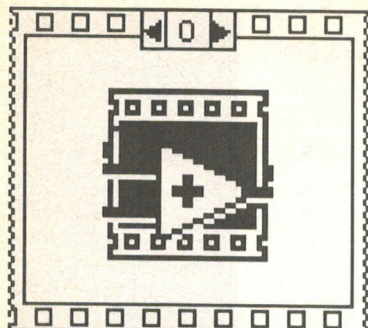
## 16ビットマシンへの移行(PC9801の出現)

PC9801が登場して8ビットマシンに別れを告げるときが来ても，CP/Mの遺産から離れられない人が多かったのですが，メガソフトのMX-1(多くのCP/Mのディスクフォーマットをサポート)とカノープスのPlus-80(PC9801上で，CP/Mのソフトを走らせるハードとソフト)が，PC8801からPC9801への移行に大きな役割を果たしたと思っています．

思い起こせば，PC9801のMS-DOSの上で，初めて「VJE」(バックス)(REDというエディタは，これにバンドリングされていた)が登場したときはショックでした．当時は，パソコンでは実用的に日本語を使える環境は全くなかったのです．CP/M86にもMS-DOSにも，簡単なかな漢字変換はあったのですが，とても実用的とはいえませんでした．

当時の専用ワープロに負けない漢字変換ソフトを手にし，データベースなどの実用ソフトで使えることを知ったとき，CP/Mの遺産を大量に持っていた筆者も，迷わずCP/M86からMS-DOSに転向したのでした．このときから，言語も日本語の完全なサポートが必要になりました．

今でも，初期PC9801のMS-DOSの普及にもっとも貢献したのはVJEだと思っています．「日本語ワードプロセッサ」(「松」の前身)も一太郎も，ただのワープロでしたが，VJEは生まれながらの仮名漢字FEPだったのです．

(つづく)

# 計測制御のためのツール LabVIEW プログラミング

## LabVIEW のオプション機能を使う

計測マニア

前回は，電子天秤を例として，簡単な重量測定のプログラムを作ってみました．そこで，今度はこれに色々なオプションをくっつけて，普通のプログラムらしい顔にしてみたいと思います．

また，前回説明できなかった印字テクニックなども説明したいと思います．

### LabVIEW のウィンドウと実行オプション

#### ● ウィンドウの設定

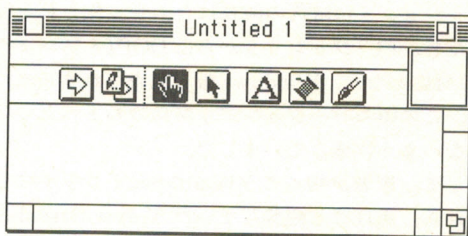
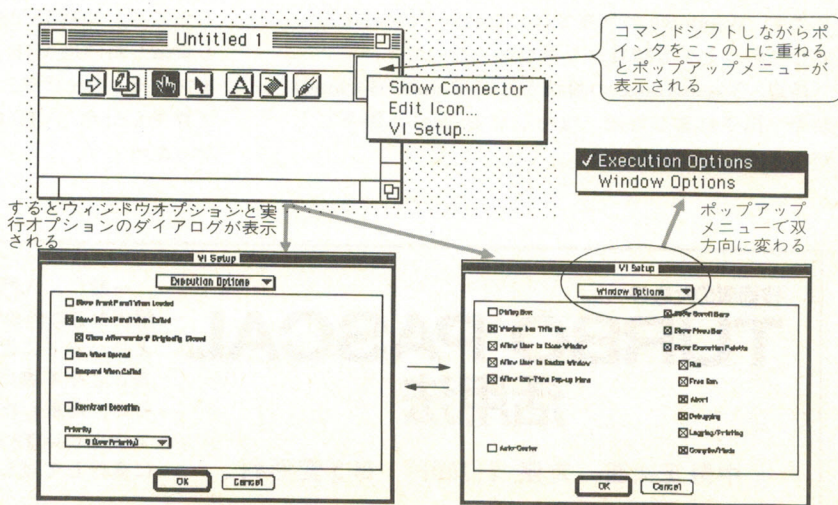
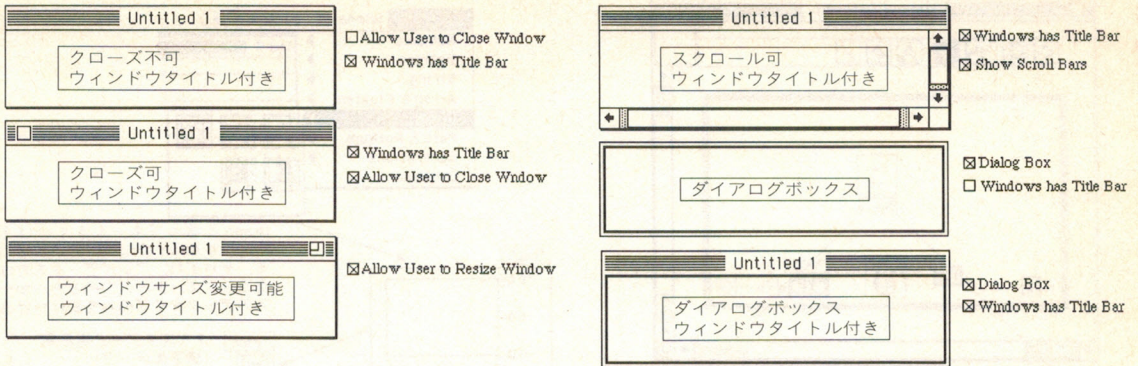
前回に作った簡単な重量測定プログラムは，特にウィンドウ関係のオプション設定を設定していませんでした．そのため，ユーザーがアプリケーションを使っているときに予期しない不具合にぶつかることが考えられます．

例えば，プログラム実行中にクローズボックスを閉じてしまったため，強制終了してしまうとか，誤ってウィンドウをスクロールしてしまい，表示が隠れてしまうなどの不具合発生が考えられます．**図5-1**がLabVIEW のデフォルトのウィンドウ設定状態です．

そこで，これらの不具合防止のためにLabVIEW に準備されているウィンドウのタイプを設定してみましょう．まず，ウィンドウのオプション設定はパネルのアイコン枠をコマンドクリックすることにより，設定ダイアログを呼び出して設定できます(**図5-2** 参照)．

ウィンドウのタイプ選択ダイアログが表示されたら自分の希望するウィンドウタイプにチェックマーク(X マーク)を入れると，スクロール不可のウィンドウやダイアログタイプのウィンドウが簡単に設定できます．

**図5-3** に，その例をいくつか示します．チェックマークの選択の組み合わせで，一般的に必要と考えられるウィンドウタイプのほぼ全てが選択できることが分かります．

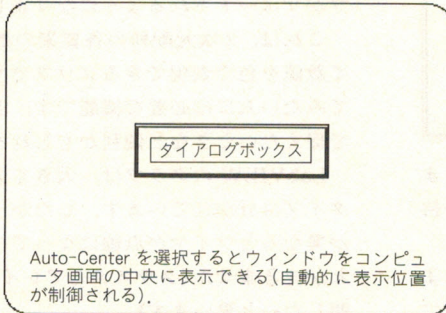
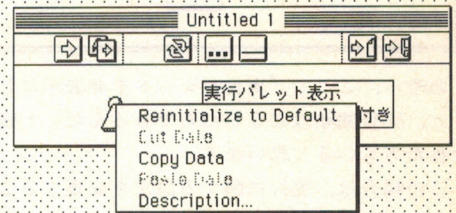
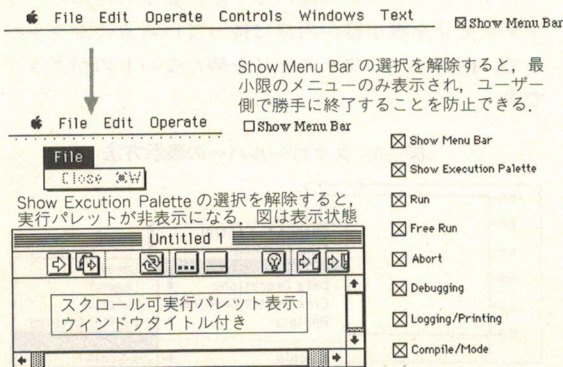
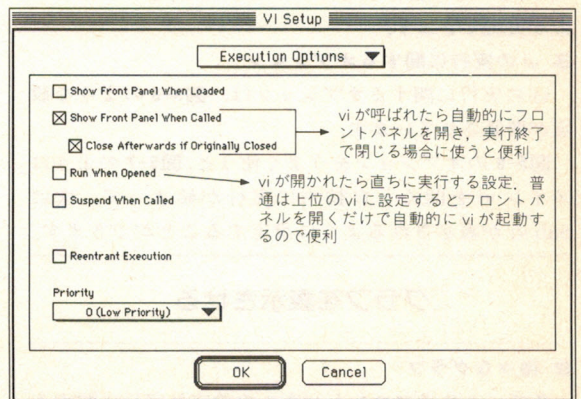
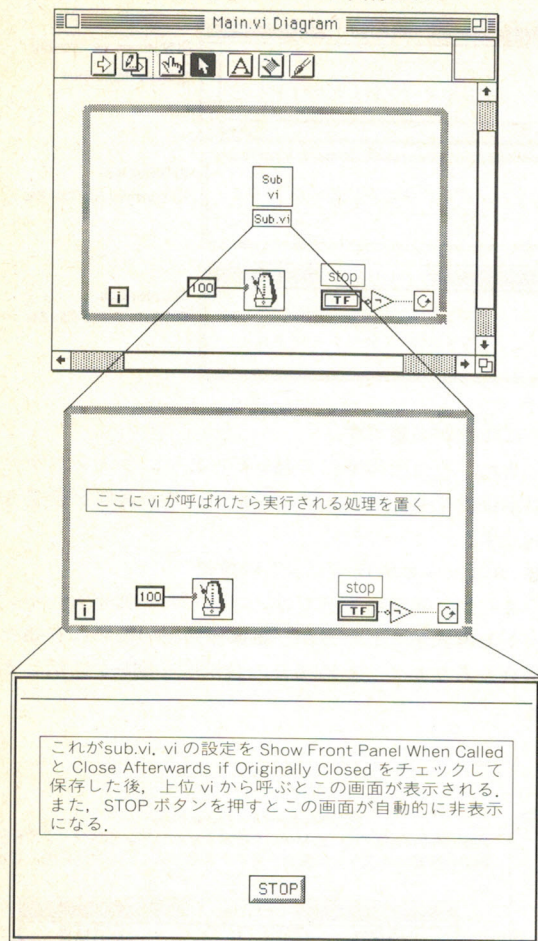
〈図5-1〉 LabVIEW のデフォルトのウィンドウ

〈図5-2〉
ウィンドウオプション設定ダイアログ

〈図5-3〉ウィンドウタイプの例

これらのウィンドウタイプは，編集モード時は反映されず，実行モード時に反映されるため，設定した後は一度実行モードを選択して確認することを忘れないでください．

また，ダイアログを選択したときは，キーボードからのコマンド+Mのキーボード・ショートカットで，実行モードを編集モードに戻すことができます．ダイアログを選択したときは，マウスからメニューが選択できなくなるので，キーボードからモード変更するように注意が必要です．

また，その他の選択で**図5-4**のようにウィンドウの表示位置やPop-upメニューについての選択が設定可能です．

### ● メニューや実行パレットの設定

さらに，場合によってはメニューの一部や実行パレットも非表示とした方が，誤操作防止のために有効な場合もあります．その場合の設定例を**図5-5**に示します．

〈図5-4〉ウィンドウの表示位置やPop-upメニューの選択方法

〈図5-5〉実行パレットを非表示にする方法

〈図5-6〉viの実行に関するオプションダイアログ

〈図5-7〉 自動的に sub.vi が実行される

おやっ，ここで「実行パレットを非表示にしてしまうと，実行開始はどうしたらできるんだ」と疑問を持たれた方もいると思います．

この場合は，実行に関する設定を変更することによって vi が実行され，上位 vi から呼ばれたときのみに下位 vi のウィンドウが表示され，実行が終わると自動的に閉じる設定を行うというテクニックを使ったり，vi が開いたら実行するオプションを設定することにより実現できます．

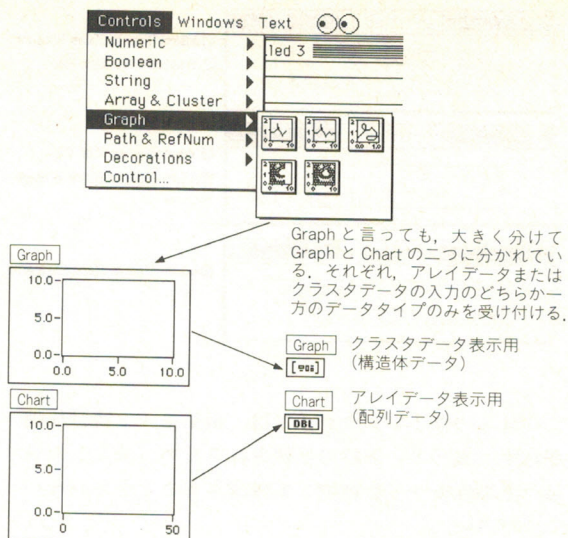
● **vi の実行に関するオプション**

vi の実行に関するオプションは，**図5-6** のような設定が準備されています．

**図5-6** のオプションをうまく使うと，**図5-7** のようにメイン vi を開くと自動的に実行が始まって，次に Sub.vi が表示されるように設定することができます．

## グラフを表示させる

● **様々なグラフ**

さて，これでフロントパネルの設定と実行に関するオプションは設定できました．次は，測定データをグラフ表示させてみましょう．LabVIEW のグラフに関するオブジェクトは，**図5-8** に示したとおりです．バージョンが 2.2 までは輝度グラフはサポートされていませんでしたが，バージョン 3 になってからは輝度グラフがサポートされるようになりました．

これは，2 次元配列の各要素の数値の大きさによって数値を色で表現できるグラフです．画像処理をやってみたい人には必要な機能です．温度分布を色の変化で捉えたいときなど便利かもしれません．

LabVIEW のグラフは，大きく分けて二つのデータタイプに分かれています．したがって，データタイプが異なるとワイヤが点線になってしまい，接続することができません(クラスタとアレイについては，後で説明したいと思います)．

〈図5-8〉 LabVIEW のグラフに関するオブジェクト

● **グラフとチャートの違い**

グラフとチャートの違いですが，表示のためのバッファメモリを表示器が内部に持っているものがチャートで，内部にバッファメモリを持たないものがグラフです．

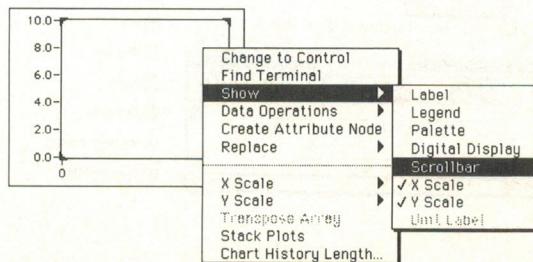
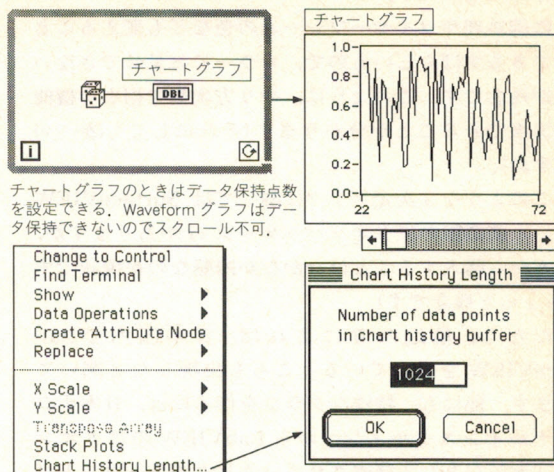
〈図5-9〉 スクロールバーの表示方法

したがって，チャートでは過去のデータを見るためのスクロールバーを持つことができますが，グラフはできません．逆に，グラフの良さはデータ表示開始ポイント(X＝0)やデータの表示間隔(dx)などを細かく設定できることがメリットです．

**図5-9** に，チャートのスクロールバーの表示方法を示します．一般的な使い分け方としては，散布図などにはグラフを使い，トレンド表示などにチャートを使うのが一般的ではないかと思います．

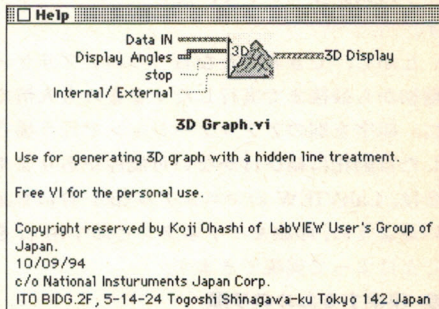
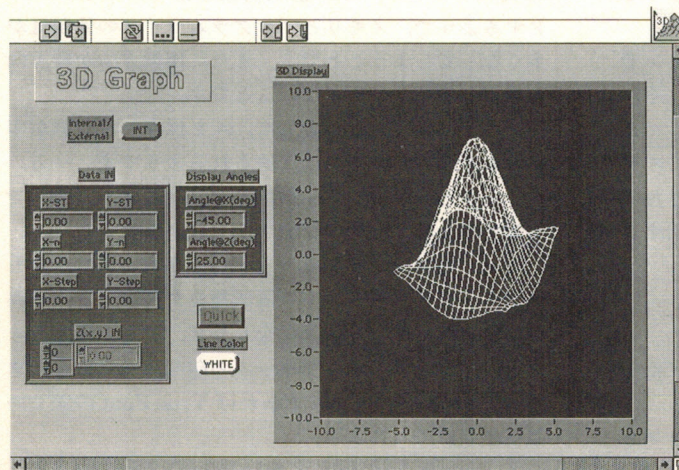
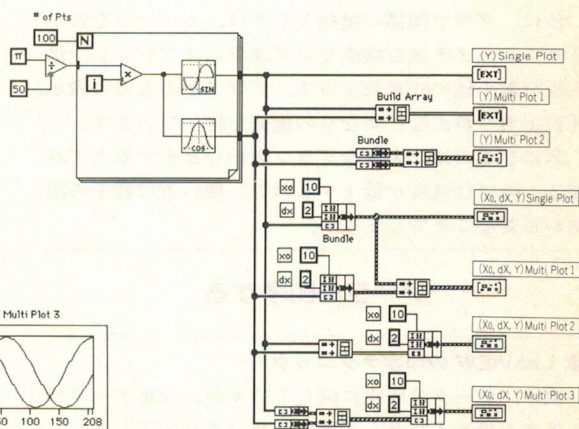
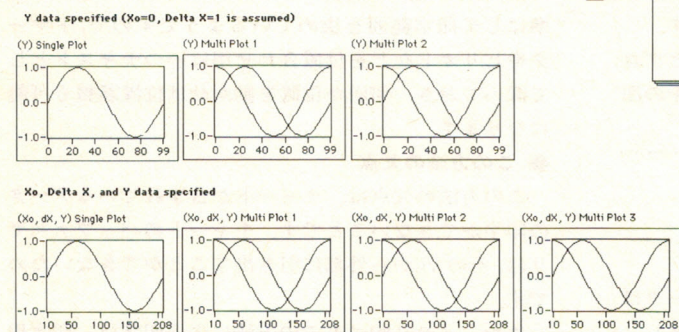
**図5-10** に，チャートのバッファメモリ変更方法を示します．

**図5-11** に示したのが，LabVIEW の example に含まれているグラフ機能を説明した vi で，Waveform Graph.vi です．

この例を見ると，いずれの場合もグラフ化したいデータは連続的に発生したアレイデータでなければならないことが分かります．一つのチャートグラフに何本かのデータを表示させたいときは，Build Array を使えばよいことがわかります．また，X＝0，dx の設定方法がわかると思います．

〈図5-10〉 チャートのバッファメモリの変更方法

### ● グラフの応用 vi

グラフに関する機能をうまく使うと，**図5-12** のような vi も作ることができます．この vi は，データを 3 次

〈図5-11〉 LabVIEW のグラフ機能を説明した vi

〈図5-12〉
データを 3 次元表示する vi

元で表示するviで，LabVIEWユーザー会の会員の方が作ったものです．

陰線処理やワイヤーフレームの色なども変えることもできるすばらしいもので，カラーでお見せできないのが残念です．このように，やり方次第で相当な機能を実現できることが分かります(それにしても凄いviですね)．

このような3次元グラフに関しては，SurfaceVIEWという商業版のアドオンパッケージviがありますが，日本から購入することはなかなか困難なのが現状です(ちょっと残念です)．

ただし，最近の噂によれば，日本国内でSurfaceVIEWを売っているところも出現したと聞いています．他にも，特殊なグラフを作るには，日本ナショナルインスツルメンツからLabVIEWのアドオンパッケージとして発売されているピクチャーコントロールキットと呼ばれているアドオンパッケージを購入することにより，かなり特殊なグラフも作ることができます．

特殊なスケールのグラフを急いで作らなければならないような場合は有効かもしれません．

他に，グラフ関係の機能としては，カーソルを表示させてデータを読む機能やログ表示のオプション，最大値や最小値の任意設定方法，グラフそのものの表示，非表示切り替えなどかなりの機能を持っています．

気になるのは，大きなグラフ表示などをするとプログラムの実行速度が低下するので，使い方に若干の注意が必要なことでしょうか．

## データを印字する

### ● LabVIEWの印字テクニック

次に，データの印字に関してですが，収集データはテキストファイルとして残されればそれでよしとし，印字は表計算ソフトで，というパターンも多いようです．

しかし，できることなら一つのアプリケーションで最初から最後まで実行したくなるのも人情です．それに，印字を別のアプリケーションで行う場合は，一貫した自動化に結び付かない可能性があります．この場合は，LabVIEWのマニュアル(p.3-4)にも記述されているように，印刷モードボタンと印字用viを活用することによって実現できます．

### ● 印字するための手順

手順としては，

(1) 上位viの印字対象の表示器をアイコン枠に接続する(**図5-13**参照)．

(2) 新しく印字用にviをオープンして，そのフロントパネルに上位viの印字対象の表示器をコピーした後，2回ペーストする(**図5-13**参照)．

(3) どちらか一方の表示器を，制御器であるControlに属性変更する．

(4) 印字用viのオブジェクトをアイコン枠に結線する．

(5) できたら制御器を全てHide Controlにしてフロントパネルから非表示にする．

(6) Fileメニューの Page Setup...を選択して，用紙方向を選んでOKとする．

(7) Fileメニューの Page Layout...を選択して，レイアウト設定用のダイアログを表示させる．

(8) ダイアログでPrint Front Panelのみをチェックして，他のチェックを全て外す．そして，OKボタンを押す．

(9) 印字用viのモードを，編集モードから実行モードにして印刷モードボタンを押す．

(10) いったん印字用viを保存する．

(11) 上位viのダイアグラムに印字ボタンを設け，設定をLatch When Releasedに選択する．

(12) 上位viのダイアグラムにケースを設け，ケース条件に印字ボタンを接続する．

(13) ケースのTrue側に印字用viを取り込む．

以上，かなり長い手順になってしまいましたが，これで印字ボタンを押すと印字ができるようになります．

後は印字位置の調整ですが，LabVIEWは左上を基準にして印字範囲を決めているようですので，＋マークやピリオドなどを位置合わせ用に一つテキストとして置いておき，何回か位置を動かせば位置調整も可能になります．

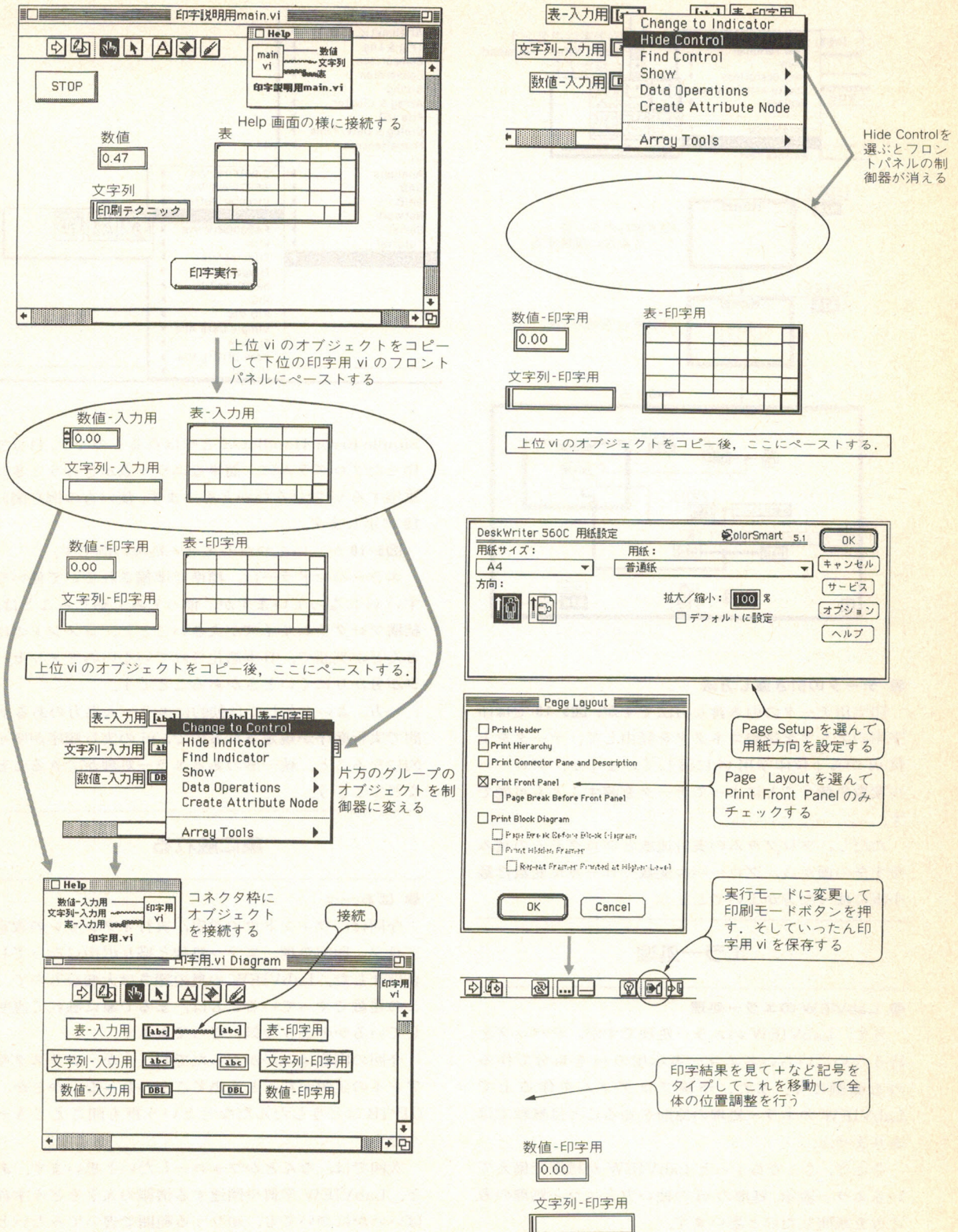
### ● この方法の欠点

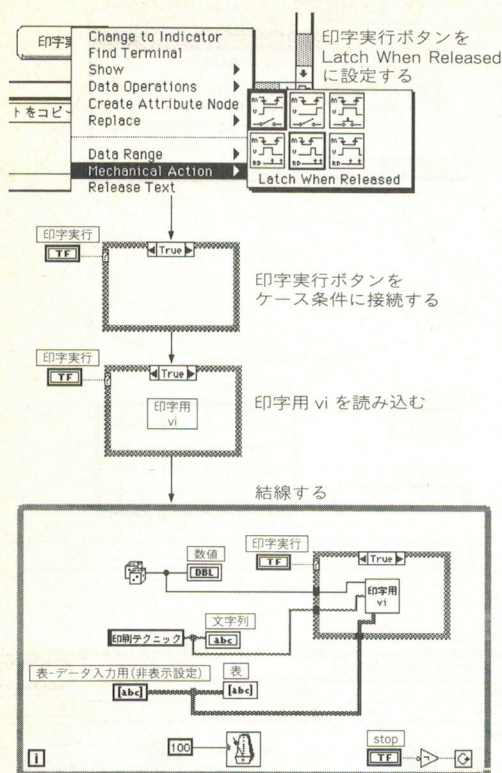
この方法の欠点は，チャートが含まれているデータの印字ができないことです．チャートのバッファメモリは，そのviから外部に引き出すことができないためです．

また，メモリ的にもこの方法による印字は，印字用のviのフロントパネル構成情報を全てメモリにロードするようで，メモリとプログラムサイズの使用量が多くなってしまうことです．また，用紙方向が縦方向と横方向の混在ができないことです．

どちらか1方向に設定して，viを保存した状態が印字するときの用紙方向に反映されるようです．一方，メリットとしては，ミリ単位のオブジェクト調整ができるので，非常に柔軟な印字が可能になることです．

データ整理用として定型フォーマットが決まったものであっても，根気よく位置調整をすれば，かなり実用に耐えうる印字が可能になります．また，印字フォントは，そのプラットフォームの印字機能に依存しますが，MacやWindowsであればTrue Fontの美しい印字結果が得られます．

〈図5-13〉データを印字するための手順

〈図5-13〉 データを印字するための手順(つづき)

〈図5-14〉 エラーハンドルのメニューの場所

● データの引き渡し方法

印字用データの引き渡し方法ですが，**図5-13**では印字用のデータを接続コネクタを経由して，データを上位 vi から下位印字用 vi に渡していますが，グローバル変数を使うことによってデータを渡すことも可能です．

ただし，プログラムの実行速度とプログラムの読みやすさの面から，グローバル変数やローカル変数は最小限に使った方がよいでしょう．

## エラー処理

● LabVIEW のエラー処理

さて，LabVIEW のエラー処理ですが，デバッグを行うためにいちいちエラー表示用の vi を自分で作るのも面倒です．また，今後プログラムを作る上で LabVIEW のエラー処理の原則を知ることは無駄ではありません．

そこで，ここでちょっと LabVIEW が標準で備えているエラー表示，処理の vi の使い方とエラー処理の考え方を説明したいと思います．

まず，エラー処理の vi は，**図5-14**の vi が標準で準備されています．最も簡単なエラーハンドラー vi は Simple Error Handler.vi と呼ばれるものです．自分で作ったプログラムに，簡単なエラー表示を行うときに重宝する vi ではないかと思います．使い方の例を**図5-15**に示します．

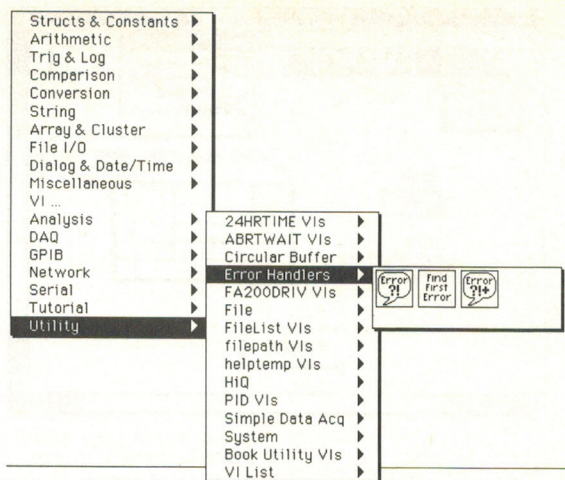



**図5-16**が，いくつかのエラー処理の例です．

エラーハンドラーは，標準で準備されていて使いやすい vi になっていますが，使ってみて感じることは，結構プログラムサイズが大きいことと，コメントのほとんどが英語で，日本語化されていないのでメッセージが分かりにくいときがあることです．

一方，よい点として自動的にエラー入出力のある vi 間で実行順序が規定されるので，vi の実行順序が明示的になること，統一感のあるエラー処理ができることなどでしょう．

## 象に触れる

● ぱお〜っ

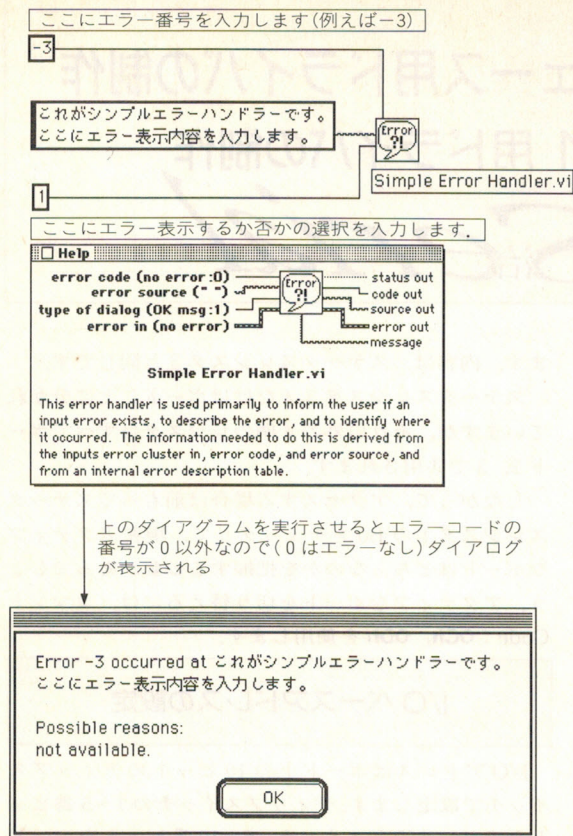
今回は，ウィンドウの設定，実行オプションの設定，グラフ，印字処理，エラー処理と盛り沢山になってしまいました．LabVIEW の奥の深さは大変なもので，この連載でやっていることは，まるで象に触れて説明しているかのような気になってきます．

今回の内容はいかがだったでしょうか．クラスタやアレイの説明がちっとも出てこないじゃないかとか，GP-IB はどうしたんだなどという声も聞こえてきそうです．

次回では，なんとかフォローしたいと思います．あと，LabVIEW 学習や関連する情報の入手をどうすればいいかについても，知りうる範囲で書いてみたいと思います．

最近，ある国内計測器メーカーは新製品の計測器に

〈図5-15〉 Simple Error Handler の使い方

〈図5-16〉 エラー処理の例

は標準的に LabVIEW のドライバ vi を開発，配付できるように体制を整えつつあります．それぱかりか，過去に販売した計測器のドライバについても徐々に整備して行くそうです．LabVIEW のユーザーにとっては喜ばしいことです．(^_^)

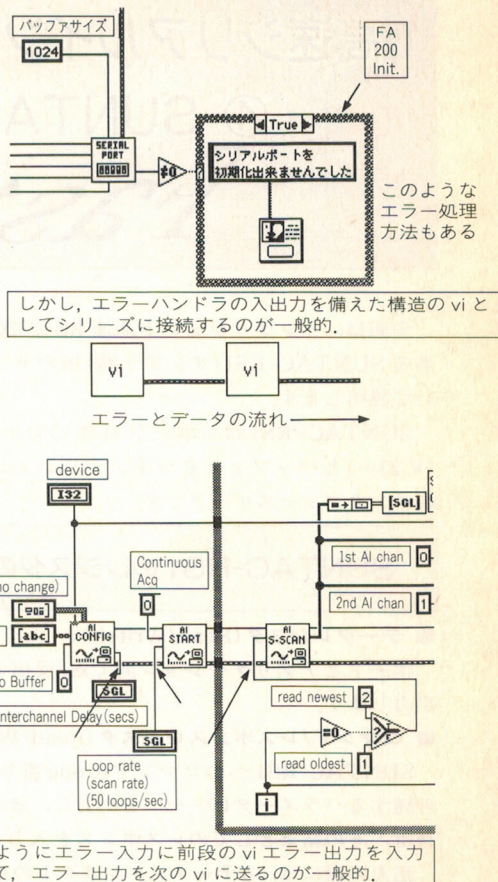
◆ 参考文献 ◆


(1) 多田眞作；第4回　グラフィック言語による計測プログラミング，トラ技コンピュータ 1993年8月号.

(2) 田辺敏也；LabVIEW を用いた計測機器とパソコンの接続の実際，インターフェース 1994年12月号.

(3) LabVIEW マニュアル一式，日本ナショナルインスツルメンツ㈱.

(4) SEAM92' CD-ROM, MacSciTech, 1992, 国内では LaboFinder(☎ 0425-92-7767)より入手可能.

(5) 井上泰典；実践 LabVIEW 3 入門，ソニーテクトロニクス㈱システム営業部.

(6) G.W.Johnson, LabVIEW Graphical Programming Techniques, McGraw-Hill, ISBN0-07-032692-4.

(7) ElectronicsUpdate, 1994年5月号.

(8) LaboFinderNewsletter, Vol.1, No.3, June 1992.

(9) G.W.Johnson, LabVIEW with Style, MacSciTech's Data Acquisition Image Acquisition Data Analysis CD-ROM, MacSciTech, 1993.

(10) John P Kerrigan, 小西宏明；Macintosh for Expert LabVIEW, ㈱BNN, ISBN4-89369-344-1.

(11) 日本ナショナルインスツルメンツ㈱，LabVIEW 基本コースマニュアル，部品番号 320751-01.

(12) 大橋康司；LabVIEW による有限要素解析，LaboFinder Newsletter Vol.3, No 4, 1995年.

(13) 大橋康司；3D Graph.vi, LabVIEW User's Group of Japan Newsletter 第3号, 1995年.

# 高速シリアルインターフェース用ドライバの制作
# ④ SUNTAC-RS1 用ドライバの制作

Fast Serial

石川 治

今回は，インテリジェント高速 RS-232C ボードである SUNTAC-RS1(サン電子㈱)用のドライバについて説明します．

SUNTAC-RS1 は，ボード自体にローカル CPU (V 25＋)とバッファをもつインテリジェントタイプのインタフェースボードです．

## SUNTAC-RS1 のレジスタの概要

● **データレジスタ** (Read/Write)

ボードとアプリケーション間で送/受信データを入出力します．

● **コマンド/レスポンスレジスタ** (Read/Write)

SUNTAC-RS1 へのコマンド(Code 番号)やそれに付随するパラメータ(データ)を出力し，また必要な場合は，その結果としてのレスポンスを入力します．

出力の際には，ステータスレジスタ 1 のビット#5 が 0 になるのを確認して次の動作に移ります．入力の際には，ステータスレジスタ 1 のビット#4 が 1 になるのを確認してから読み出す必要があります．

● **ステータスレジスタ 1** (Read only)

ローカル CPU のステータス情報などを取得します(**図4-1**)．ブロック転送モードの場合には，1 バイトのデータごとに B2，B3 を確認する必要はないようですが，ホスト側が高速な場合については定かではありません．割り込みを使用しない場合は，B1 の確認は必要ありません．

● **ステータスレジスタ 2** (Read only)

ボードのバッファのステータス情報などを取得します(**図4-2**)．B0，B1，B2 は，割り込み送受信の場合に確認が必要になるフラグです．

● **ステータスレジスタ 3** (Read only)

ポート 2 の RS-232C ラインの信号の状態を取得します(**図4-3**)．DCE 側からの信号のみではなく，DTE 側からの信号も確認できます．信号線と同じくすべて負論理であることに注意してください．

● **ステータスレジスタ 4** (Read only)

ポート 3 の RS-232C ラインの信号の状態を取得します．内容は，ステータスレジスタ 3 と同じです．

ステータスレジスタ 3, 4 だけはポートごとに分かれていますが，それ以外の 4 個のレジスタはすべてポート 2，3 で共用されます．

したがって，アクセスする場合は前もってステータスレジスタ 1 の B0 をチェックして，現在アクティブなポートはどちらなのかを把握する必要があるでしょう．アクティブなポートを切り替えるには，コマンド Code：**0Ch**，**0Dh** を使用します．

## I/O ベースアドレスの設定

I/O アドレスはボード上の 10 ビットのディップスイッチで設定します．ディップスイッチの 1〜5 番ビットをアドレス設定スイッチ，6〜10 番ビットをボード番号設定スイッチとして説明されています．

I/O アドレスとディップスイッチの各ビットの関係は，**表4-1** のようになっています．

## SUNTAC-RS1 のコマンドの概要

ここでは今回のドライバで使用するコマンドについて特筆すべきことのみコメントすることにとどめます(**表4-2**)．その他の詳細についてはマニュアルを参照してください．

ほとんどのコマンドは，発行した時点でアクティブなポートに対してのみ有効ですが，一部のコマンドは両方のポートに同時に作用するので注意してください．

## 制作したドライバの特色

**① 割り込み制御は一切使わない**

このボードは一応割り込み制御にも対応はしているものの，基本的な設計思想として，ポーリング制御で動作させることを想定しているようです．

したがって，今回の SUNTAC-RS1 用ドライバでは割り込みを一切使いません．UART 用のドライバに比べるとはるかにコンパクトに仕上がります．

なお割り込みを使用しないので，ボード上の割り込みレベルを選択するためのジャンパピンは必ず抜いてください(出荷時の設定は **INTO** になっている).

**② ドライバ自体にはバッファは持たない**

①と関連して，これもインテリジェントボードであることの大きなメリットのひとつです．ドライバは，アプリケーションとボードとの間で，片方から受け取

〈図4-1〉 ステータスレジスタ 1
(ローカルCPUのステータス情報など)

〈図4-2〉 ステータスレジスタ 2
(ボードのバッファのステータス情報など)

(1) B0，B1，B2 は，割り込み送/受信の場合に確認が必要になるフラグ．

(2) B6 は，ROM RS 2.41 になって新たに追加されたフラグ．現在コマンドを処理中の場合このビットが1となり，新規コマンドを発行しても無視される．新たにコマンドを発行する場合には，このビットが0であることを確認する必要がある(それ以前のバージョンでは，常に0)．

〈図4-3〉 ステータスレジスタ 3
(ポート2のRS-232Cラインの信号の状態)

〈表4-1〉 I/Oアドレスとディップスイッチの各ビットの関係

| I/Oアドレス | A15 | A14 | A13 | A12 | A11 | A10 | A09 | A08 | A07 | A06 | A05 | A04 | A03 | A02 | A01 | A00 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| DipSW・No | 10 | 9 | 8 | 7 | 6 | ** | ** | ** | *1 | *1 | 5 | 4 | 3 | 2 | 1 | *0 |

ボード番号ビット (A15～A11)　　I/Oアドレスビット (A05～A01)

**の3ビットでレジスタをセレクト(0～5)
*1 は1に固定
*0 は0に固定

| レジスタ | I/Oアドレス | レジスタ | I/Oアドレス |
|---|---|---|---|
| データレジスタ | x0nnh | ステータスレジスタ 2 | x3nnh |
| コマンド/レスポンスレジスタ | x1nnh | ステータスレジスタ 3 | x4nnh |
| ステータスレジスタ 1 | x2nnh | ステータスレジスタ 4 | x5nnh |

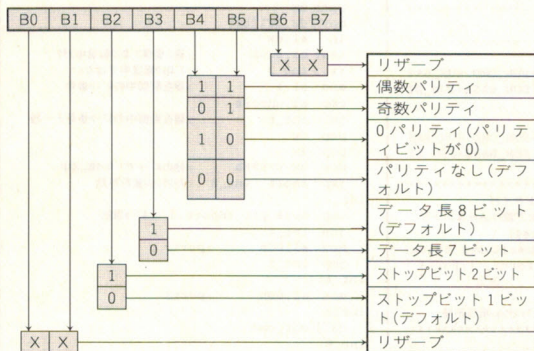
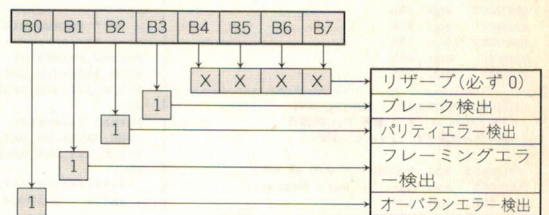
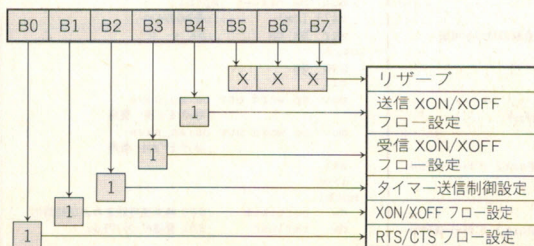
x 部分は，ボード番号ビットで決まる
nn 部分は，I/Oアドレスビットで決まる

〈表4-2〉SUNTAC-RS1のコマンド(◆：今回使用する　◇：使用しない)

| コード(Code) | 意味 | コメント |
|---|---|---|
| **00h** ◇ | 割り込み発生受信データサイズ指定 | 受信割り込みを使用する場合に必要 |
| **01h** ◆ | 送信ブロック転送サイズ指定(DMA転送) | データとしてブロック送信したい希望バイト数を入力すると，レスポンスとしてブロック転送の可能サイズ(≦バッファの空きサイズ)が返される(最大255バイト).<br>このあと，ステータスレジスタ1のビット#7(送信ブロック転送中フラグ)がONに転ずるのを確認して，レスポンス分だけ送信データを出力するループを繰り返すことになる.<br>送信ブロック転送では，ステータスレジスタ1のビット#2(送信データラッチ状態フラグ)をいちいち確認する必要はない |
| **02h** ◆ | 受信ブロック転送サイズ指定(DMA転送) | 受信ブロック転送は，ローカルCPUによる受信データの自動ポート出力を禁止にした状態のままで行う必要がある(Code：**1Dh**).<br>データとしてブロック受信したい希望バイト数(通常はアプリケーション側の作業用バッファのサイズ)を入力すると，レスポンスとしてブロック転送可能なサイズ(≦有効な受信データ数)が返される(最大255バイト).<br>このあと，ステータスレジスタ1のビット#6(受信ブロック転送中フラグ)がONに転ずるのを確認して，レスポンス分だけ受信データを入力するループを繰り返す.<br>なお，受信ブロック転送では，ステータスレジスタ1のビット#3(データレジスタに受信データ有りフラグ)を確認する必要はない |
| **03h** ◆ | 送信バッファクリア要求 | バッファがクリアされるだけ．データもレスポンスもない |
| **04h** ◆ | 受信バッファクリア要求 | **03h**と同じ．データもレスポンスもない |
| **05h** ◆ | 通信プロトコル設定 | 非同期通信モードの通信パラメータを指定する(**図4-4**) |
| **06h** ◇ | フロー制御要求 | XON/XOFF信号を最優先に送信するコマンド．そのほかとして，ブレーク制御も行う．ブレーク信号制御はCode：**12h**にも用意されているが，このコマンドでは自動的に300msのブレーク信号を出すのでむしろ使いやすいかもしれない |
| **07h** ◇ | タイマー送信間隔指定 | 遅延送信のためのコマンド．Code：**11h**でタイマー送信制御が設定された場合に有効 |
| **08h** ◆ | ラインエラー・ステータス要求 | データビットの内容は**図4-5**のとおり．エラーフラグは，エラークリア・コマンドが発行されるまで保持される |
| **09h** ◇ | モデムステータス要求 | モデムステータス(RS-232C信号線の状態)は，ステータスレジスタ3，または4から直接読み込むことにする.<br>このコマンドはモデムステータスに変化があったかどうかを知りたい場合に有用 |
| **0Ah** ◆ | 端末速度選択 | 与えるデータと設定される端末速度との関係は**表4-3**のとおり |
| **0Bh** ◇ | プロトコル通信選択(未対応) | |
| **0Ch** ◆ | COM2ポート選択(デフォルト) | **0Ch**，**0Dh**により，アクティブなポートを切り替える．二つのポートを同時に駆動する場合，頻繁に発行することになるが，どちらかのポートがDMA動作中である場合にはポート切り替えを保留するのが得策.<br>またポート切り替えは，受信データの自動ポート出力を禁止して行う必要がある |
| **0Dh** ◆ | COM3ポート選択 | |
| **0Eh** ◇ | 割り込み要因選択 | 各種の割り込み条件を許可/禁止する |
| **0Fh** ◇ | 割り込み要因ステータス取得 | 割り込みが発生したとき，その割り込み要因のステータスを取得する |
| **10h** ◇ | 割り込み解除 | 割り込み要因をクリアする |
| **11h** ◆ | フロー制御指定 | データビットの内容は**図4-6**のとおり |
| **12h** ◇ | ブレーク信号強制制御 | ビット#0の1/0でブレーク信号をON/OFFする |
| **13h** ◇ | ボードインターフェース要求 | ボードとアプリケーションとの間でのエラーの有無を取得する |
| **14h** ◆ | エラークリア要求 | Code：**08h**の説明にもあるように，エラーフラグはこのコマンドを発行しないとクリアされない |
| **15h** ◆ | 送信バッファ空きサイズ取得要求 | 送信バッファに空きがあるかどうかの確認に使用するが，レスポンスとして実際の空きバイト数より1バイト大きい値を返してくるので注意が必要.<br>ブロック送信のバイト数は，Code：**01h**のレスポンスを用いるのが正確 |
| **16h** ◆ | 受信バッファ有効サイズ取得要求 | 有効な受信データのバイト数を知ることができる．ブロック受信でのバイト数は，やはりCode：**02h**のレスポンスを用いる |
| **17h** ◇ | ボード情報取得要求 | ROMのバージョンなどの情報を得る |
| **18h** ◇ | 送受信許可設定 | 送受信動作の許可/禁止を設定するものだが，デフォルトのまま(ともに許可)で使用 |
| **19h** ◇ | ダイアログモード移行要求 | |

〈表4-2〉 SUNTAC-RS1のコマンド(◆：今回使用する　◇：使用しない，つづき)

| コード(Code) | 意　味 | コメント |
|---|---|---|
| **1Ah** ◆ | ボード初期化要求 | ドライバの常駐時に一度だけ実行する．動作は電源投入時と同じなので，ポート2，3両方とも初期化される．<br>多少時間がかかるがフラグの確認が必要．まず，ステータスレジスタ2のビットがすべて0にクリアされるのを待ち，その後，ブロック転送要求フラグが許可になるのを待って先に進む．<br>ドライバの立ち上げ時，他のボードと違って注意しなければならないのは，この点と，ゴミデータ/ゴミレスポンスの除去 |
| **1Bh** ◆ | DTR信号ON要求 | DTR(ER)信号をONにする |
| **1Ch** ◆ | DTR信号OFF要求 | DTR(ER)信号をOFFにする |
| **1Dh** ◆ | 受信データの自動ポート出力禁止 | 受信をブロック転送で行う場合には，CPUの自動ポート出力を禁止しておかなければならない．今回のドライバでは，両ポートとも常に禁止状態のままで動作させることにする |
| **1Eh** ◇ | 受信データの自動ポート出力許可(デフォルト) | Code：**1Dh**で自動ポート出力を禁止にしたポートを，許可に戻すコマンド，今回のドライバは基本的に受信ブロック転送(DMA動作)のままで動かすので，許可にしてはいけない |
| **1Fh** ◆ | 受信データの1バイト出力要求 | 受信データの自動ポート出力禁止状態で，受信ブロック転送によらずに1バイトだけ受信バッファからデータを取り出すときに必要になる |
| **20h** ◇ | RTS信号ON要求 | これらはフロー制御をRTS/CTSに設定していない場合についてのみ有効． |
| **21h** ◇ | RTS信号OFF要求 | |
| **22h** ◆ | 送信バッファサイズの指定 | ボード上には12Kバイトの送信バッファを持っているが，なんらかの事情で送信をキャンセルしたいという場合，それでは大きすぎるということもある．<br>このコマンドで256バイト単位でデータを指定することで，送信バッファのサイズを任意の大きさに変更可能．今回のドライバも，送信バッファは2Kバイトに縮小している．なお小さくした結果，バッファの残った部分は切り捨てることになる |
| **23h** ◇ | ポート別割り込み要因選択 | |
| **24h** ◇ | ポート別割り込み要因ステータス取得 | |

〈図4-4〉 コマンド05h(通信プロトコル設定)

〈図4-5〉 コマンド08h(ラインエラー・ステータス要求)

〈図4-6〉 コマンド11h(フロー制御指定)

〈表4-3〉 データと端末速度の関係

| データ | 端末速度(bps) | データ | 端末速度(bps) |
|---|---|---|---|
| 15(**0Fh**) | 153600 | 7(**07h**) | 19200 |
| 14(**0Eh**) | 115200 | 6(**06h**) | 14400 |
| 13(**0Dh**) | 76800 | 5(**05h**) | 9600 |
| 12(**0Ch**) | 62400 | 4(**04h**) | 7200 |
| 11(**0Bh**) | 57600 | 3(**03h**) | 4800 |
| 10(**0Ah**) | 41600 | 2(**02h**) | 2400 |
| 9(**09h**) | 38400 | 1(**01h**) | 1200 |
| 8(**08h**) | 20800 | 0(**00h**) | リザーブ |

ったデータをもう片方に即座に渡すだけ．つまり，中継するだけです．

**③ ドライバ立ち上げ時に，一度だけボードの初期化作業を行う**

電源投入時にゴミデータやゴミレスポンスが入る可能性があるということで，それらのゴミを除去する必要があります．Code：**1Ah** のボード初期化コマンドも発行します．

**④ ポートパラメータエリアについて**

ステータスレジスタ 3,4 を除いて他のすべてのレジスタはふたつのポートで同じものを共用しますので，ポートパラメータを二つのポートにあえて分ける必要もありませんが，他のボードのドライバと仕様を整えるために一応エリアを分けました．

## ドライバ各部の解説

ハードウェア割り込み処理部がありませんので，ドライバは，

> ソフトウェア割り込み処理部
> 常駐処理部

の二つの部分から構成されます(**リスト4-1**)．

## ソフトウェア割り込み処理部

### ● エントリルーチン

アプリケーション側からコールされる割り込みベクター番号には **19h** を使用します．

自前のスタックを用意し，AH レジスタの内容からポート番号，ファンクション番号を得てジャンプするのは UART の場合と同じです．

ただし，SUNTAC-RS1 の場合は二つのポートで一つのレジスタを共用するという制約があるため，対応する番号のポートをエントリルーチンでアクティブにしておく必要があります．具体的には，Code：**0Ch**, **0Dh** のポート選択コマンドを発行することです．

最初に現在ブロック転送中であるかどうかを確認し，転送中で，かつそれが他方のポートで動作中であれば，ポート切り替えはできません．ファンクションの実行不能のままアプリケーションに戻ります(このとき，BX＝**FFFFh** とする)．

それ以外の場合は，設定された番号に対応するポートを選択してジャンプします．ファンクション実行後，BX レジスタをゼロクリアして戻ります．

ファンクション：**0Ch** のキャリア確認のみ，各ポートで独立したステータスレジスタを読みに行くのでポ

〈リスト4-1〉 SUNTAC-RS1 用シリアルドライバ **SUNTAC.ASM**

```
; SUNTAC−RS1用ドライバ   for ROM RS2.41
; 1995.07.14  Copyrigyt (C) O.Ishikawa

title  suntac.asm

PUTSTR    equ  09h
SETVECT   equ  25h
KEEPRC    equ  31h
GETVECT   equ  35h
MEMFRE    equ  49h
ENDPRC    equ  4Ch
;-----------------------------------------
;SUN ELECTORONICS  SUNTAC-RS1
;  下位I/Oｱﾄﾞﾚｽ, ﾎﾞｰﾄﾞ番号 DipSW選択
;  割り込み使用せず (ｼﾞｬﾝﾊﾟﾋﾟﾝは抜く)

SUN_BASE equ  00D0h  ;ﾍﾞｰｽ･ｱﾄﾞﾚｽ ｾｯﾄ
SUN_DAT  equ  0h     ;Data Register
SUN_CMD  equ  100h
              ;Command/Response Register
SUN_ST1  equ  200h ;Status Register 1
SUN_ST2  equ  300h ;Status Register 2
SUN_ST3  equ  400h ;Status Register 3
SUN_ST4  equ  500h ;Status Register 4
;-----------------------------------------
Parameter   struc
  DAT     dw   ? ;Data Register
  CMD     dw   ? ;Command/Response Register
  ST1     dw   ? ;Status Register 1
  ST2     dw   ? ;Status Register 2
  ST3     dw   ? ;Status Register 3/4
Parameter   ends

code   segment
       assume   cs:code, ds:code
       org  80h
;=========================================
; コマンドライン・パラメータ   [/R] option
;=========================================
cmdln  db   128 dup(?)
       org  100h
;=========================================
; ドライバ本体
;=========================================
start: jmp  main
ofs19     dw   ?
seg19     dw   ?

onport    db   ? ;動作中のﾎﾟｰﾄ:0:Port2,1:Port3
ss_save   dw   ? ;ｽﾀｯｸ･ｾｸﾞﾒﾝﾄ
sp_save   dw   ? ;ｽﾀｯｸ･ﾎﾟｲﾝﾀｰ
          db   256 dup (?)
                     ;INT19 での一時的ｽﾀｯｸ領域
mystack  label  byte
;----------------------------------------
; Port Parameters Area
;----------------------------------------
Param2 Parameter
<SUN_BASE+SUN_DAT,SUN_BASE+SUN_CMD,SUN_BAS
E+SUN_ST1,SUN_BASE+SUN_ST2,SUN_BASE+SUN_ST
3>
Param3 Parameter
<SUN_BASE+SUN_DAT,SUN_BASE+SUN_CMD,SUN_BAS
E+SUN_ST1,SUN_BASE+SUN_ST2,SUN_BASE+SUN_ST
4>
;****************************************
; INT19 : シリアルインタフェースドライバ
;                ・ソフトウェア割り込みエントリ
; in :AH  ChannelNumber(bit#4)
;           + Function Number(Lower 4bits)
;     Channel  Number:  0:Port2, 1:Port3
;     Funtion  Number:  0h .. 0Ch
; out:BX  = 0FFh
;       他のﾎﾟｰﾄでﾌﾞﾛｯｸ転送中につきﾌｧﾝｸｼｮﾝ実行不能
;****************************************
int19_entry:
  mov  word ptr cs:ss_save,ss
                          ;ｽﾀｯｸ･ｾｸﾞﾒﾝﾄ 退避
  mov  word ptr cs:sp_save,sp
                          ;ｽﾀｯｸ･ﾎﾟｲﾝﾀｰ 退避
  mov  bx,cs
  mov  ss,bx         ;
  mov  sp,offset mystack  ;自前のｽﾀｯｸを用意
  sti
  push ds
  push cs
  pop  ds
  test ah,10h        ;bit#4 をﾁｪｯｸ
  jnz  int_1
  mov  byte ptr onport,0
  mov  si,offset Param2   ;Port2 ﾊﾟﾗﾒｰﾀ･OFFｾｯﾄ
  jmp  short  int_2
int_1:
  mov  byte ptr onport,1
  mov  si,offset Param3   ;Port3 ﾊﾟﾗﾒｰﾀ･OFFｾｯﾄ

int_2:
  mov  bl,ah
  and  bx,000Fh      ;ﾌｧﾝｸｼｮﾝ番号 (下位4bit)
  cmp  bl,0Ch        ;getdcd ?
  je   int_7         ;ﾎﾟｰﾄ選択不要
  push ax
  push dx
  mov  dx,ST1[si]
  in   al,dx
  test al,0C0h           ;送/受信ﾌﾞﾛｯｸ転送中ﾌﾗｸﾞ
  jz   int_3             ;ﾌﾞﾛｯｸ転送中ではない
  and  al,1              ;現在制御中のﾎﾟｰﾄ番号
  cmp  al,onport
  je   int_6             ;現在制御中のﾎﾟｰﾄ番号と一致
  pop  dx
  pop  ax
  mov  bx,0FFFFh         ;他のﾎﾟｰﾄでﾌﾞﾛｯｸ転送中
  jmp  short  int_8  ;ﾌｧﾝｸｼｮﾝ実行不能
int_3:
  cmp  byte ptr onport,0  ;ﾎﾟｰﾄ選択
  jne  int_4
  mov  al,0Ch       ;port2
  jmp  int_5
int_4:
  mov  al,0Dh       ;port3
int_5:
  call out_cmd
int_6:
  pop  dx
  pop  ax
int_7:
  shl  bx,1
  add  bx,offset jmptbl
  call [bx]         ;各ﾌｧﾝｸｼｮﾝへ
  xor  bx,bx        ;BX <- 0
int_8:
  pop  ds
  cli
  mov  sp,word ptr cs:sp_save
                    ;ｽﾀｯｸ･ﾎﾟｲﾝﾀｰ 復帰
  mov  ss,word ptr cs:ss_save
                    ;ｽﾀｯｸ･ｾｸﾞﾒﾝﾄ 復帰
  sti
  iret
jmptbl:
  dw   portinit     ;$00 端末速度設定および初期化
  dw   rxflush      ;$01 受信ﾊﾞｯﾌｧ･ﾌﾗｯｼｭ
```

ートの切り替えは不要です.

● **コマンド/パラメータ出力とレスポンス入力**

コマンド/レスポンスレジスタを介しての，コマンドおよびそれに付随するパラメータ(データ)の出力，およびそのコマンドに対するボードからのレスポンスの入力を行います.

それぞれ，ステータスレジスタ1のビット#5ないし#4を確認して入出力します.

なおROM RS2.41からステータスレジスタ2のビット#6にコマンド処理中フラグが追加されましたので，コマンド出力の際にはこのフラグにより新規コマンドの受付け禁止/許可状態の確認も必要です.

● **0h：portinit(端末速度の設定/ポート初期化)**

▶端末速度：ALレジスタの値によって設定
▶通信パラメータ：パリティなし，データ長8ビット，ストップビット1ビットで固定
▶DTR信号をON
▶フロー制御：デフォルトのRTS/CTSフロー
▶送信バッファサイズ：2Kバイトに縮小
▶受信データの自動ポート出力を禁止：アプリケーションとの送/受信は，ブロック転送(DMA動作)を基本として行うので，これは重要.
▶送/受信バッファの初期化

● **1h：rxflush(受信バッファの初期化)**

ボードの受信バッファクリア・コマンドを発行し，さらに念のためラインエラー・ステータスをクリアーしておきます.

● **2h：txflush(送信バッファの初期化)**

ボードの送信バッファクリア・コマンドを発行するだけです.

● **3h：recvbyte(受信バッファより1バイト入力)**

ボードの受信バッファ内にデータがあるのを確認し，あれば1バイトだけ読み出してALレジスタを介してアプリケーションに引き渡します.

常に受信データの自動ポート出力を禁止状態で動作させているので，バッファ内にデータがあっても，ローカルCPUは受信データを勝手にはデータレジスタに送り込めません.

先に受信データの1バイト出力要求コマンドを発行し，ステータスレジスタ1のビット#3(受信データ有りフラグ)を確認してデータを受け取ります.

● **4h：sendbyte(送信バッファへ1バイト出力)**

ボードの送信バッファに空きバイト数があるのを確認し，あればアプリケーション側からALレジスタを介して渡された1バイトのデータを，データレジスタから出力します.

出力後，ステータスレジスタ1のビット#2(送信デー

〈リスト4-1〉 SUNTAC-RS1用シリアルドライバ**SUNTAC.ASM**(つづき)

```
    dw   txflush      ;$02 送信ﾊﾞｯﾌｧ･ﾌﾗｯｼｭ
    dw   recvbyte     ;$03 1ﾊﾞｲﾄ受信
    dw   sendbyte     ;$04 1ﾊﾞｲﾄ送信
    dw   recvblk      ;$05 ﾌﾞﾛｯｸ受信
    dw   sendblk      ;$06 ﾌﾞﾛｯｸ送信
    dw   getrxbuf     ;$07 受信ﾊﾞｯﾌｧ内ﾃﾞｰﾀ数取得
    dw   gettxbuf     ;$08 送信ﾊﾞｯﾌｧ空き領域ｻｲｽﾞ取得
    dw   getstat      ;$09 ﾎﾟｰﾄのｽﾃｰﾀｽ取得
    dw   setctrl      ;$0A ﾎﾟｰﾄ･ｺﾝﾄﾛｰﾙ
    dw   setflow      ;$0B ﾌﾛｰ制御の設定
    dw   getdcd       ;$0C ｷｬﾘｱの有無の確認
    dw   dummy        ;$0D
    dw   dummy        ;$0E
    dw   dummy        ;$0F
dummy:
    ret
;----------------------------------------
; コマンド出力     in :AL ｺﾏﾝﾄﾞ番号
;----------------------------------------
out_cmd:
    push ax
    mov  dx,ST1[si]       ;ｽﾃｰﾀｽ･ﾚｼﾞｽﾀ･1
wait00:
    in   al,dx
    and  al,20h           ;ｺﾏﾝﾄﾞ･ﾗｯﾁ･ﾌﾗｸﾞ
    jnz  wait00
    mov  dx,ST2[si]       ;ｽﾃｰﾀｽ･ﾚｼﾞｽﾀ･2
wait01:
    in   al,dx
    and  al,40h           ;ｺﾏﾝﾄﾞ処理中ﾌﾗｸﾞ
    jnz  wait01
    pop  ax
    mov  dx,CMD[si]       ;ｺﾏﾝﾄﾞ/ﾚｽﾎﾟﾝｽ･ﾚｼﾞｽﾀ
    out  dx,al            ;RS1 <-- AL(ｺﾏﾝﾄﾞ)
    ret
;----------------------------------------
; パラメータ出力     in :AL ｺﾏﾝﾄﾞﾊﾟﾗﾒｰﾀ
;----------------------------------------
out_prm:
    push ax
    mov  dx,ST1[si]       ;ｽﾃｰﾀｽ･ﾚｼﾞｽﾀ･1
wait1:
    in   al,dx
    and  al,20h           ;ｺﾏﾝﾄﾞ･ﾗｯﾁ･ﾌﾗｸﾞ
    jnz  wait1
    pop  ax
    mov  dx,CMD[si]       ;ｺﾏﾝﾄﾞ/ﾚｽﾎﾟﾝｽ･ﾚｼﾞｽﾀ
    out  dx,al            ;RS1 <-- AL(ﾊﾟﾗﾒｰﾀ)
    ret
;----------------------------------------
; レスポンス入力   out:AL ﾚｽﾎﾟﾝｽ
;----------------------------------------
in_res:
    mov  dx,ST1[si]       ;ｽﾃｰﾀｽ･ﾚｼﾞｽﾀ･1
wait2:
    in   al,dx
    and  al,10h           ;ﾚｽﾎﾟﾝｽ･ｱﾘ･ﾌﾗｸﾞ
    jz   wait2
    mov  dx,CMD[si]       ;ｺﾏﾝﾄﾞ/ﾚｽﾎﾟﾝｽ･ﾚｼﾞｽﾀ
    in   al,dx               ;AL <-- RS1(ﾚｽﾎﾟﾝｽ)
    ret
;****************************************
; $00 : 端末速度の設定
;          およびチャンネル初期化  (SUNTAC-RS1)
; in :AL (=端末速度ﾃﾞｰﾀ 1..15)
;    1:1200;2:2400;3:4800;4:7200;
;    5:9600;6:14400;7:19200;8:20800;
;    9:38400;10:41600;11:57600;12:62400;
;    13:76800;14:115200;15:153600;
; out:none
;****************************************
portinit:
    push ax
    mov  al,0Ah         ;端末速度 設定
    call out_cmd
    pop  ax
    call out_prm
    mov  al,05h         ;通信ﾊﾟﾗﾒｰﾀ 設定
    call out_cmd
    mov  al,08h
                 ;NonParity,8bit長,Stop1bit
    call out_prm
    mov  al,1Bh           ;DTR信号 ON
    call out_cmd
    mov  al,11h           ;ﾌﾛｰ制御 設定
    call out_cmd
    mov  al,01h           ;RTS/CTS ﾌﾛｰ (ﾃﾞﾌｫﾙﾄ)
    call out_prm
    mov  al,22h           ;送信ﾊﾞｯﾌｧ･ｻｲｽﾞ 設定
    call out_cmd
    mov  al,8h            ;256*8 = 2Kﾊﾞｲﾄ
    call out_prm
    mov  al,1Dh           ;受信ﾃﾞｰﾀ自動ﾎﾟｰﾄ出力 禁止
    call out_cmd
    call txflush          ;送信ﾊﾞｯﾌｧ 初期化
    call rxflush          ;受信ﾊﾞｯﾌｧ 初期化
    ret
;****************************************
; $01 : 受信バッファの初期化
;        in :none
;        out:none
;****************************************
rxflush:
    mov  al,04h           ;受信ﾊﾞｯﾌｧ ｸﾘｱ
    call out_cmd
    mov  al,14h           ;ﾗｲﾝ･ｴﾗｰ･ｽﾃｰﾀｽ ｸﾘｱ
    call out_cmd
    ret
;****************************************
; $02 : 送信バッファの初期化
;        in :none
;        out:none
;****************************************
txflush:
    mov  al,03h           ;送信ﾊﾞｯﾌｧ ｸﾘｱ
    call out_cmd
    ret
;****************************************
; $03 : 受信バッファより1文字入力
;        in :none
;        out:AL=Data
;****************************************
recvbyte:
    call getrxbuf
    and  ax,ax
    jnz  l030
    ret                 ;受信ﾊﾞｯﾌｧにﾃﾞｰﾀがない
l030:
    mov  al,1Fh         ;受信ﾃﾞｰﾀ 1ﾊﾞｲﾄ出力要求
    call out_cmd        ;ﾃﾞｰﾀ･ﾚｼﾞｽﾀ <- 受信ﾊﾞｯﾌｧ
    mov  dx,ST1[si]     ;ｽﾃｰﾀｽ･ﾚｼﾞｽﾀ･1
l031:
    in   al,dx
    and  al,08h           ;ﾃﾞｰﾀ･ｱﾘ･ﾌﾗｸﾞ
    jz   l031
    mov  dx,DAT[si]       ;ﾃﾞｰﾀ･ﾚｼﾞｽﾀ
    in   al,dx            ;受信ﾃﾞｰﾀ入力
    xor  ah,ah
    ret
;****************************************
; $04 : 送信バッファへ1文字出力
```

タラッチフラグ)を見て，ローカルCPUが確実にデータを受け取ったか確認しておきます．

送信バッファに送られたデータは，後はボードの送信制御に委ねられて順次送出されることになります．

● **5h：recvblk**(受信バッファよりブロック転送)

受信バッファ内のデータを，CXレジスタで指定されたバイト数だけまとめてアプリケーションに引き渡します．

このときボード内のバッファからデータレジスタへのデータ転送にはローカルCPUを介さずにDMA転送で行われます．ローカルCPUを介してのブロック転送に比較するとかなり速いことが確認されています．

また，1バイトデータの授受のたびにステータスレジスタ1のビット#3を確認する必要もありません．

▶動作内容

まずステータスレジスタ2のビット#4で受信ブロック転送要求が許可であるのを確認してから受信ブロック転送要求コマンド(Code：**02h**)を発行します．

続けてブロック転送したいバイト数(≦255)を出力します．それに対してレスポンスとしてブロック転送可能なバイト数(≦希望バイト数)が返されます．

可能バイト数が0でなければ，ボード側は自動的にDMA転送モードに移行するので，それをステータスレジスタ1のビット#6(受信ブロック転送中フラグ)で確認します．

ONに転じたなら直ちに可能バイト数だけデータレジスタから入力するループに入ります．

入力ループ終了後，もう一度ステータスレジスタ1のビット#6をチェックし，今度はそれがONからOFFに転じたのを確認して終わります．

● **6h：sendblk**(送信バッファへブロック転送)

アプリケーションからの送信データを，CXレジスタで指定されたバイト数だけまとめて送信バッファに引き渡します．

このときデータ転送にはローカルCPUを介さずにDMA転送で行われるのはブロック受信と同じです．1バイトデータの授受のたびにステータスレジスタ1のビット#2を確認する必要もありません．

▶動作内容

まず，ステータスレジスタ2のビット#5で送信ブロック転送要求が許可であるのを確認してから送信ブロック転送要求コマンド(Code：**01h**)を発行します．

続けてブロック転送したいバイト数(≦255)を出力します．それに対してレスポンスとしてブロック転送可能なバイト数(≦希望バイト数)が返されます．

なおこの可能バイト数と，Code：**15h**を用いる**gettxbuf**で得られる値とは必ずしも一致しません．送信バッファの空きサイズとしては，ここで得られる値

〈リスト4-1〉SUNTAC-RS1用シリアルドライバ**SUNTAC.ASM**(つづき)

```
;       in :AL=Data
;       out:none
;*****************************************
sendbyte:
  mov  cl,al            ;CL <- 送信ﾃﾞｰﾀ
  call gettxbuf         ;送信ﾊﾞｯﾌｧ 空きﾊﾞｲﾄ数
  and  ax,ax
  jnz  1040
  ret                   ;送信ﾊﾞｯﾌｧに空きがない
1040:
  mov  dx,DAT[si]       ;ﾃﾞｰﾀ・ﾚｼﾞｽﾀ
  mov  al,cl
  out  dx,al            ;送信ﾃﾞｰﾀ出力
  mov  dx,ST1[si]       ;ｽﾃｰﾀｽ・ﾚｼﾞｽﾀ・1
1041:
  in   al,dx
  and  al,04h           ;送信ﾃﾞｰﾀ・ﾗｯﾁ・ﾌﾗｸﾞ
  jnz  1041
  ret
;*****************************************
; $05 : 受信バッファよりブロック転送
; in :ES:DI (=top address),
;                  CX (=length >0)
; out:AL : 転送ﾊﾞｲﾄ数; AH=FFh : 転送失敗
;*****************************************
recvblk:
  call getrxbuf
  and  ax,ax
  jnz  1050
  ret                 ;受信ﾊﾞｯﾌｧにﾃﾞｰﾀがない
1050: cmp  ax,cx
  jae  1051
  mov  cx,ax
1051:
  mov  dx,ST2[si]       ;ｽﾃｰﾀｽ・ﾚｼﾞｽﾀ・2
  in   al,dx
  and  al,10h           ;受信ﾌﾞﾛｯｸ転送要求許可ﾌﾗｸﾞ
  jnz  1052
  mov  ax,0FF01h        ;受信ﾌﾞﾛｯｸ転送禁止
  ret
1052:
  mov  al,02h           ;ﾌﾞﾛｯｸ受信転送ｻｲｽﾞ
  call out_cmd
  mov  al,cl            ;AL <- ﾌﾞﾛｯｸ転送希望ｻｲｽﾞ
  call out_prm
  call in_res           ;AL <- ﾌﾞﾛｯｸ転送可能ｻｲｽﾞ
  xor  ah,ah            ;AH <- 0
  and  ax,ax
  jnz  1053
  ret                       ;ﾌﾞﾛｯｸ転送不可 (AX=0)
1053:
  push ax
  mov  cx,ax
  mov  bx,0
  mov  dx,ST1[si]       ;ｽﾃｰﾀｽ・ﾚｼﾞｽﾀ・1
1054:
  in   al,dx
  and  al,40h           ;受信ﾌﾞﾛｯｸ転送中ﾌﾗｸﾞ
  jnz  1055
  dec  bx
  jnz  1054
  pop  ax
  mov  ax,0FF02h        ;受信ﾌﾞﾛｯｸ転送ｴﾗｰ
  ret
1055:
  mov  dx,DAT[si]   ;ﾃﾞｰﾀ・ﾚｼﾞｽﾀ
1056:
  in   al,dx            ;受信ﾃﾞｰﾀ入力
  stosb                 ;ES:[DI] <- AL , inc DI
  loop 1056             ;ﾌﾞﾛｯｸ転送のﾙｰﾌﾟ
  mov  bx,0
  mov  dx,ST1[si]   ;ｽﾃｰﾀｽ・ﾚｼﾞｽﾀ・1
1057:
  in   al,dx
  and  al,40h           ;受信ﾌﾞﾛｯｸ転送中ﾌﾗｸﾞ
  jz   1058
  dec  bx
  jnz  1057
  pop  ax
  mov  ax,0FF03h    ;受信ﾌﾞﾛｯｸ転送ｴﾗｰ
  ret
1058:
  pop  ax               ;ﾌﾞﾛｯｸ転送ﾊﾞｲﾄ数
  ret
;*****************************************
; $06 : 送信バッファへブロック転送
; in :ES:DI (=top address), CX (=length)
; out:AL : 転送ﾊﾞｲﾄ数; AH=FFh : 転送失敗
;*****************************************
sendblk:
  and  cx,cx
  jnz  1060
  xor  ax,ax
  ret
1060:
  call gettxbuf         ;送信ﾊﾞｯﾌｧ 空きﾊﾞｲﾄ数
  and  ax,ax
  jnz  1061
  ret                   ;送信ﾊﾞｯﾌｧに空きがない
1061:
  cmp  ax,cx
  jae  1062
  mov  cx,ax
1062:
  mov  dx,ST2[si]   ;ｽﾃｰﾀｽ・ﾚｼﾞｽﾀ・2
  in   al,dx
  and  al,20h           ;送信ﾌﾞﾛｯｸ転送要求許可ﾌﾗｸﾞ
  jnz  1063
  mov  ax,0FF01h    ;送信ﾌﾞﾛｯｸ転送禁止
  ret
1063:
  mov  al,01h           ;ﾌﾞﾛｯｸ送信転送ｻｲｽﾞ
  call out_cmd
  mov  al,cl            ;AL <- ﾌﾞﾛｯｸ転送希望ｻｲｽﾞ
  call out_prm
  call in_res           ;AL <- ﾌﾞﾛｯｸ転送可能ｻｲｽﾞ
  xor  ah,ah            ;AH <- 0
  and  ax,ax
  jnz  1064
  ret                   ;ﾌﾞﾛｯｸ転送不可
1064:
  push ax
  mov  cx,ax

  mov  bx,0
  mov  dx,ST1[si]           ;ｽﾃｰﾀｽ・ﾚｼﾞｽﾀ・1
1065:
  in   al,dx
  and  al,80h               ;送信ﾌﾞﾛｯｸ転送中ﾌﾗｸﾞ
  jnz  1066
  dec  bx
  jnz  1065
  pop  ax
  mov  ax,0FF02h            ;送信ﾌﾞﾛｯｸ転送ｴﾗｰ
  ret
1066:
  mov  dx,DAT[si]           ;ﾃﾞｰﾀ・ﾚｼﾞｽﾀ
1067:
  mov  al,es:[di]
  inc  di
  out  dx,al                ;送信ﾃﾞｰﾀ出力
```

が正しいということに注意してください.

この後受信ブロック転送と同様に，可能バイト数が0でなければボード側は自動的にDMA転送モードに移行します.

ステータスレジスタ1のビット#7(送信ブロック転送中フラグ)がONに転じたなら直ちに可能バイト数だけデータレジスタから出力するループに入ります.

出力ループ終了後もう一度ステータスレジスタ1のビット#7をチェックし，それがONからOFFに転じたのを確認して終わります.

● **7h：getrxbuf**(受信バッファに入っているバイト数の取得)

受信バッファ内の有効な受信データのバイト数を取得します.

● **8h：gettxbuf**(送信バッファの空き領域バイト数の取得)

送信バッファの有効な空き領域バイト数を取得します. バッファとデータレジスタの中間に，1バイト溜めておく部分があってそれもカウントするらしく，実際の送信バッファの空きバイト数より1バイト大きい値が戻ってきますので注意してください.

● **9h：getstat**(ポートの受信エラー状況の取得)

受信時のエラー状況を取得します. 各ビットの意味は，コマンド(Code：**08h**)の項を参照してください.

なお，このコマンドを発行しただけではエラーフラグはクリアされません. エラーステータス取得後は，エラークリア要求コマンドを発行してエラーステータスをクリアしておきます.

● **Ah：setctrl**(ポートのコマンドの設定)

RS-232Cラインの制御を行うコマンドですが，実際にはALレジスタで引き渡したデータによってDTR信号をON/OFFするのみです.

RTS信号のON/OFFコマンドも別にありますが，これらはフロー制御の設定がRTS/CTSフローでない場合にのみコントロール可能ということなので，今回のドライバではあえて使用しません.

● **Bh：setflow**(フローコントロールの設定)

他のボード用のドライバに合わせて，ALレジスタのビット#0でXON/XOFFフロー，ビット#1でRTS/CTSフローをそれぞれ設定することとしたため，ちょっぴり煩雑になってしまいました.

● **Ch：getdcd**(DCD信号の確認)

RS-232CラインのDCD(キャリア検出)信号を確認し，ONのときだけALレジスタの値を0にして戻ります. 負論理であることに注意してください.

このコマンドだけは，要求されたポートに応じてそれぞれ別々に用意されているステータスレジスタ3，4を直接読みにいくので，アクティブポートの変更は必

〈リスト4-1〉 SUNTAC-RS1用シリアルドライバ**SUNTAC.ASM**(つづき)

```
  loop 1067             ;ﾌﾞﾛｯｸ転送のﾙｰﾌﾟ
  mov  bx,0
  mov  dx,ST1[si]         ;ｽﾃｰﾀｽ･ﾚｼﾞｽﾀ･1
1068:
  in   al,dx
  and  al,80h           ;送信ﾌﾞﾛｯｸ転送中ﾌﾗｸﾞ
  jz   1069
  dec  bx
  jnz  1068
  pop  ax
  mov  ax,0FF03h        ;送信ﾌﾞﾛｯｸ転送ｴﾗｰ
  ret
1069:
  pop  ax               ;ﾌﾞﾛｯｸ転送ﾊﾞｲﾄ数
  ret

;****************************************
; $07 : 受信バッファに入っているバイト数の取得
;       in :none
;       out:AX
;****************************************
getrxbuf:
  mov  al,16h           ;受信ﾊﾞｯﾌｧ･ﾊﾞｲﾄ数
  call out_cmd
  call in_res           ;下位ﾊﾞｲﾄ
  mov  ah,al
  call in_res           ;上位ﾊﾞｲﾄ
  xchg al,ah
  ret
;****************************************
; $08 : 送信バッファの空き領域バイト数の取得
;       in :none
;       out:AX
;****************************************
gettxbuf:
  mov  al,15h           ;送信ﾊﾞｯﾌｧ空きｻｲｽﾞ
  call out_cmd
  call in_res           ;下位ﾊﾞｲﾄ
  mov  ah,al
  call in_res           ;上位ﾊﾞｲﾄ
  xchg al,ah
  ret
;****************************************
; $09 : ポートの受信エラー状況を与える
;       in :none
;       out:AL=Error Status
;           #3:Break, #2:PE, #1:FE, #0:OE
;****************************************
getstat:
  mov  al,08h           ;ﾗｲﾝ･ｽﾃｰﾀｽ･ﾘｰﾄﾞ
  call out_cmd
  call in_res           ;AL: ﾚｽﾎﾟﾝｽ
  mov  ah,al
  mov  al,14h           ;ﾗｲﾝ･ｴﾗｰ･ｽﾃｰﾀｽ･ｸﾘｱ
  call out_cmd
  mov  al,ah
  ret
;****************************************
; $0A : ポートのコマンドを設定する
; in :AL=MODEM Control Data
;       (0=DTR OFF; 1=DTR ON)
; out:none
;****************************************
setctrl:
  test al,01h
  jz   set5
  mov  al,1Bh           ;DTR信号 ON
  jmp  short  set6
set5:
  mov  al,1Ch           ;DTR信号 OFF
set6:
  call out_cmd
  ret
;****************************************
; $0B : フローコントロール設定
; in :AL
;    (bit#0: XON/XOFF ﾌﾛｰ ｾｯﾄ  1=ON , 0=OFF)
;    (bit#1: RTS(RS)  ﾌﾛｰ ｾｯﾄ  1=ON , 0=OFF)
;     out:none
;****************************************
setflow:
  push ax
  call txflush
  call rxflush
  pop  ax
  mov  ah,al
  mov  al,11h           ;ﾌﾛｰ制御 設定
  call out_cmd
  clc              ;ｷｬﾘｰ･ﾌﾗｸﾞ･ﾘｾｯﾄ
  rcr  ah,1           ;Rotate Right Through
Carry
  jnb  rsf1           ;ｷｬﾘｰ･ﾌﾗｸﾞ:0 で ｼﾞｬﾝﾌﾟ
  or   ah,02h           ;ｷｬﾘｰ･ﾌﾗｸﾞ:1 で bit#1 on
rsf1:
  mov  al,ah
  call out_prm
  ret
;****************************************
; $0C : モデムステータスのCD信号を読み取る
;       in :none
;       out:AL=0 CD ON ; AL<>0 CD OFF
;****************************************
getdcd:
  mov  dx,ST3[si]
  in   al,dx
  and  al,08h           ;CD? (負論理)
  ret
;----------------------------------------
int_end     label  word
int_length = int_end - int19_entry
;========================================
; ドライバの組み込み (常駐)
;========================================
main:
  cmp  cmdln,0
  jne  p0
  jmp  short  s0
p0:
  mov  cl,0
  mov  si,offset cmdln[1]
p1:
  inc  si
  mov  al,[si]
  cmp  al,0Dh
  jne  p2
  jmp  short  s0
p2:
  cmp  al,'/'
  je   p3
  cmp  al,'-'
  jne  p1
p3:
  inc  si
  mov  al,[si]
  cmp  al,0Dh
  jne  p4
  jmp  short  s0
;++++++++++++++++++++++++++++++++++++++++
; オプション [/R] の処理
;++++++++++++++++++++++++++++++++++++++++
p4:
```

要がありません．

## 常駐処理部

基本的には前号UART用ドライバの場合と同じく，

(1) コマンドライン・オプションを読んで解釈(**/R**で常駐解除のみ)

(2) すでに常駐しているかどうかを判定

(3) その結果から

常駐している場合………そのまま常駐する．

常駐している場合………常駐を解除して終了する．

常駐していない場合……新たに常駐して終了する．

の3通りに分かれます．

ハードウェア割り込み処理のベクターの設定/解除が省かれるので，この部分もコンパクトになります．

### ● 立ち上げ時の初期化作業

ドライバの立ち上げ時(電源投入時)に初期化作業をしておく必要があります．まず，SUNTAC-RS1が拡張スロットに装着されているかどうかを確認します．

これには，ステータスレジスタ1のI/Oポートアドレスから入力した値が，**FFh**であるかどうかで判定可能です．

SUNTAC-RS1が装着されているのを確認後，ゴミデータ，ゴミレスポンスの除去を行います．まず，ステータスレジスタ1のビット#4を見てレスポンス有りビットが1である間コマンド/レスポンスレジスタからダミー入力を繰り返します．

それが済んだ後，続いてビット#3を見てデータありビットが1である間データレジスタからダミー入力します．こうしてゴミデータ/レスポンスを除去したあと，ボード初期化要求コマンド(Code：**1Ah**)を発行します．

## テスト用サンプルプログラムについて

### ● 端末速度に関して

SUNTAC-RS1のひとつだけのポートを用いての対向転送では115200 bpsでも十分可能です．

しかし二つのポートを同時に用いての自己ループ転送では76800 bpsまでが限度のようです．

115200 bps以上ではポートの切り替え動作が追従しないらしく，正常に動作しませんでした．

### ● ファームウェアのバージョンに関して

一部の古いバージョンのファームウェアではポート切り替えの際1 ms程度のウェイトを挿入しなければ化けたりハングアップしたりすることがありました．ROM RS 2.3 C以降のバージョンではその点は改善され，ウェイトなしでも問題はありません．

〈リスト4-1〉 SUNTAC-RS1用シリアルドライバ**SUNTAC.ASM**(つづき)

```
   cmp  al,'R'
   je   p5
   cmp  al,'r'
   je   p5
   jmp  short  p3
p5:
   mov  byte ptr erase,0FFh
;++++++++++++++++++++++++++++++++++++++++
;  すでに常駐しているかどうかの判定
;++++++++++++++++++++++++++++++++++++++++
s0:
   mov  ah,GETVECT
   mov  al,19h
   int  21h            ;割り込みﾍﾞｸﾄﾙ取得 -> ES:BX
   mov  word ptr ofs19,bx     ;保存
   mov  word ptr seg19,es
   mov  di,bx
   mov  si,offset int19_entry
   mov  cx,int_length
   repz cmpsb     ;DS:[SI]=ES:[DI]?
   jne  s1        ;一致しないﾃﾞｰﾀあり
;;;  すでに常駐している場合の処理
   cmp  byte ptr erase,0FFh      ;常駐解除ﾌﾗｸﾞ
   je   esc1
;++++++++++++++++++++++++++++++++++++++++
;  常駐したまま終了
;++++++++++++++++++++++++++++++++++++++++
   call suntac_init;
   mov  dx,offset msg0
   mov  ah,PUTSTR
   int  21h
   mov  ah,ENDPRC
   int  21h
;++++++++++++++++++++++++++++++++++++++++
;  常駐解除して終了
;++++++++++++++++++++++++++++++++++++++++
esc1:  cli
   push ds
   mov  dx,word ptr es:ofs19
   mov  ds,word ptr es:seg19
   mov  ah,SETVECT
   mov  al,19h
   int  21h              ;割り込みﾍﾞｸﾄﾙ 元に戻す
   pop  ds
   push es
   mov  ax,es:[002ch]
   mov  es,ax
   mov  ah,MEMFRE       ;環境のﾒﾓﾘ解放
   int  21h
   pop  es
   mov  ah,MEMFRE       ;本体のﾒﾓﾘ解放
   int  21h
   mov  dx,offset msg1
   mov  ah,PUTSTR
   int  21h
   mov  ah,ENDPRC
   int  21h
;++++++++++++++++++++++++++++++++++++++++
;  新たに常駐して終了
;++++++++++++++++++++++++++++++++++++++++
s1:
   cli
   call suntac_init;
   mov  dx,offset int19_entry
   mov  ah,SETVECT
   mov  al,19h
   int  21h           ;割り込みﾍﾞｸﾄﾙ設定
   mov  dx,offset msg0
   mov  ah,PUTSTR
   int  21h
   mov  dx,offset int_end+15
   mov  cx,4
   shr  dx,cl
   mov  ah,KEEPRC
   int  21h
;========================================
;;;  SUNTAC−RS1  初期化
suntac_init proc
   mov  si,offset Param2 ;Port2 ﾊﾟﾗﾒｯﾀ･OFFｾｯﾄ
   mov  dx,ST1[si]
   in   al,dx
   cmp  al,0FFh
   jne  si0
   mov  dx,offset msg2   ;装着されていない
   mov  ah,PUTSTR
   int  21h
   mov  ah,ENDPRC
   int  21h
si0:
   mov  dx,ST1[si]         ;ｺﾞﾐ･ﾚｽﾎﾟﾝｽの除去
   in   al,dx
   and  al,10h             ;ﾚｽﾎﾟﾝｽ･ｱﾘ･ﾌﾗｸﾞ
   jz   si1
   mov  dx,CMD[si]         ;ｺﾏﾝﾄﾞ/ﾚｽﾎﾟﾝｽ･ﾚｼﾞｽﾀ
   in   al,dx
   jmp  short  si0
si1:
   mov  dx,ST1[si]         ;ｺﾞﾐ･ﾃﾞｰﾀの除去
   in   al,dx
   and  al,08h             ;ﾃﾞｰﾀ･ｱﾘ･ﾌﾗｸﾞ
   jz   si2
   mov  dx,DAT[si]         ;ﾃﾞｰﾀ･ﾚｼﾞｽﾀ
   in   al,dx
   jmp  short  si1
si2:
   mov  al,1Ah             ;ﾎﾞｰﾄﾞ初期化要求
   mov  dx,CMD[si]         ;ｺﾏﾝﾄﾞ/ﾚｽﾎﾟﾝｽ･ﾚｼﾞｽﾀ
   out  dx,al
   mov  dx,ST2[si]         ;ｽﾃｰﾀｽ･ﾚｼﾞｽﾀ･2
si3:
   in   al,dx
   and  al,al
   jnz  si3
si4:
   in   al,dx
   and  al,30h
   cmp  al,30h
   jne  si4
   ret
suntac_init endp
;========================================
erase:
   db   0          ;常駐解除ﾌﾗｸﾞ  0FFh で解除
msg0:
   db   0Dh,0Ah,'SUNTAC-Driver 常駐しました...
1995.07.14',0Dh,0Ah,'$'
msg1:
   db   0Dh,0Ah,'SUNTAC-Driver 常駐解除しまし
た.',0Dh,0Ah,'$'
msg2:
   db 0Dh,0Ah,'[SUNTAC-RS1] が装着されていませ
ん.',0Dh,0Ah,'$'

code ends
   end  start
```

## From U.S.A. ③ 過剰機能と“Simple is best”

最近，筆者が常々感じていることの一つに，マイコン制御の家電製品を米国ではあまり目にすることがないということがあります．

筆者もかつては日本にいたことがありますが，そのころはファジー制御の洗濯機や掃除機，プログラムの豊富な電子レンジなどの宣伝をよく見ていました．しかし，こちらではそのような家庭電化製品，いわゆる白物(シロモノ)にマイコンが入っているものが非常に少ないように思います．

**★ 掃除機について**

まず，掃除機には完全にマイコンは使われていないようです．理由は簡単で，強力ならそれで十分だからです．日本の電気掃除機のように，ごみの状態を察知してパワーをコントロールするよりも，常にフルパワー動作していたほうが作りも簡単だし，使う人も安直に使えます．

それに，掃除機の場合は日本と米国では完全にその用途が異なっています．日本では畳のゴミを取ることに主眼が置かれているのに対し，米国ではカーペットの汚れとりが主な用途です．

まず，クリーナーで汚れを取って，その泡ごと吸い取り，それでおしまいっていうのがこちらの掃除機の使われ方で，きめ細かさよりも強力なことが絶対条件です．

中には，自分でお湯をカーペットに吹き掛けて汚れを吸い取るタイプもあり，ここまでくると日本の掃除機と米国のクリーナーは別ジャンルの家電と考えたほうがいいかもしれません．

日本の掃除機は，「ゴミの吸い取りに使うだけの補助的な」バキュームクリーナという感じでしょうか．

ちなみに，このタイプの掃除機(空気を吸ってついでにゴミも吸い取るタイプ)は米国では単に「バキューム」と呼ばれています．

**★ 洗濯機と乾燥機について**

洗濯機(ウォッシャ)も強力＋お湯を使うのがデフォルトで，洗剤も酵素なんか入っていません．

熱いお湯と強力な洗剤でガンガンかき回せば大抵の汚れは落ちてしまいます．軟弱な服はすぐ痛んでしまいますが，痛んだらそれでおしまいです．非常に単純明快ではありますが，それで困るようなことはまったくありません(筆者の場合，あまりいい服なんか持っていないのでまったく問題ない)．

ついでに，洗濯物を日向で干すことがこちらでは全くなく，乾燥機(ドライア)でガンガン乾かしてそれでおしまいです．

そのため，乾燥機の中に洗濯物と一緒に入れて柔軟仕上げをする紙(柔軟剤が染み込んでいる紙，乾燥機の熱風で発散して洗濯物に柔軟仕上げを施す)がこちらではよく使われます．

汚れの種類が違うということもあるでしょうが，絶対的な強さの前では姑息な回りくどさなんか太刀打ちできません(まるで日米の政府間交渉と同じですね)．

**★ 電子レンジについて**

電子レンジ，こちらではMicrowave Ovenとか，単に「マイクロウェーブ」と呼んでいますが，それについているプログラムをきちんと使っているユーザーを見たことありますか？ 筆者はありませんでした．ワット数を把握しておいて，時間を正しく設定できればそれでユーザーとしては100点満点でしょう．

その他にも，マイコン制御なんかまったくいらない家電製品は数多くあるのですが，筆者の考えるところではやはり日本のマイコン制御の家電製品の多くは，日本の伝統である「過剰機能」の精神が反映されている典型的な分野だと思います．

ちょっと冷静に考えてみましょう．そのような家電製品のユーザーはだれでしょう．

答えは，電気に対する知識のほとんどない主婦です．米国のコンシューマ産業の健全なところは，このようなユーザーにとって，まったく関係のない項目を飾り立てることが全くないことでしょう．

ちなみに，米国では，PL法が強力なため，家電製品でいいかげんな設定による誤動作があった場合，メーカを提訴することが一般化しています．

そのため，マニュアルも極めて分かりやすくかかれており，電気の知識のない一般家庭の主婦が確実に使えるようになっています．

**★ 生活の質と製品の質**

筆者が思うに，今の日本は“Simple is best. Complicated is worst”ということをもう一度再考し，生活の質と製品の質とのアンバランスを改善するべきではないでしょうか．

最後に，もうちょっとシリアスなことを言わせてもらえば，家電に限らず日本では，どうでもいいような部分にハイテク気分で人と金と時間を未だに浪費している感じがします．

蓋を開けてみれば，大抵の場合にはその会社の内製品の4ビットか8ビットマイコンが使われているのがほとんどです．

もしも読者の皆さんの中に，そういう仕事に携わっている人がいるならば，明日からPowerPCとかColdFireの組み込みアプリケーションの開発をUnixホストのVxWorksでやれとか，Socketを使っていたアプリをSTREAMSに書き直せとか，やるべきことはほかにもあるはずなのです．

さて次回は，いよいよ！ 内外から要望の多かった!? 米国カリフォルニア州サンタクララにあるIntelオフィスの敷地内にあるビーチバレー場の写真入りで，シリコンバレーのオフィスの雰囲気についてお話ししましょう．

〈Macorney E. Saitmayer〉

新連載

パソコンでUnix

# 理工系ユーザーのためのLinux入門

## [1] Linuxの導入(前編)

伊藤　敏

32ビットのオペレーティングシステム(OS)の時代といわれ，多くのOSが出てきています．この連載では，その中で，PC-Unixの一つであるLinuxを取りあげます．LinuxはJEというグループの大変な努力により英語環境(通常の環境)をインストールしたら，それに上書きをする形で日本語環境を入れると，すぐに日本語が使えるように整備されています．

はじめに，筆者の立場を紹介しましょう．インターネットにアクセスできません．Linux(PC-Unix)を使い出してからあまり経っていません．しかし，DOS, MS-Windowsをアプリケーションユーザーとして使い仕事をしてきた人間が，Unixの豊富なアプリケーションを使うためにLinuxを導入し，悪戦苦闘をした経験を持っています．この経験をもとに，できるだけ失敗を少なく導入する方法を書くことはできます．これからLinuxをやってみようと考えている方は，これを機会に，ぜひトライしてみてください．

本稿では理工系技術者が解決しなければならない問題を考え，計算し，その結果を検討して，プレゼンテーションしていくための道具としてLinuxが使えるような状況を考え，導入方法を紹介していきます．

そして，「考え，その考えを検討する」過程で道具としてコンピュータを用いる場合，マルチタスクで使えることが重要であると考えます．そこでマルチタスクを快適に使うためにX Windowの導入を前提として話を進めます．

〈写真1-1〉InfoMagic Linux CD-ROM

### 1．Linuxの導入にあたって

#### 1-1．Linuxとは，そして入手方法

LinuxはLinus B.Torvales氏が開発したPC/AT互換機で動作するUnixライクなオペレーティングシステムです．ソースとバイナリを自由に再配布できます．基本的なコマンドやツールのバイナリを合わせ，インストーラを付けたパッケージが用意されています．

ネットワークを通じていろいろなパッケージが配布されています．しかし，はじめに述べたように筆者はインターネットにアクセスできません．したがって，CD-ROMを入手して利用することになります．

アクセスできない人が利用しやすいパッケージとして，SlackwareとSLS(Softlanding Linux System)があります．いずれもディレクトリが1枚のフロッピディスクに納まるようになっています．最近は雑誌の付録CD-ROMとして配布されたり，Linux上で日本語が使えるようにパッケージ(JE)を用意するなど活躍しているグループの監修によるCD-ROMが出ています．

一番のお薦めはJEによるLinux+JEです．Laser5(東京都千代田区外神田2-1-13 細野ビル ☎03-5296-0670 通信販売可)という店がCD-ROMにして販売しています．その他にInfoMagic Linux(**写真1-1**)というCD-ROMが販売されています．これも同じ店で入手できます．

#### 1-2．必要なハードウェア

Linuxは2Mバイトのメモリ，フロッピは1台で動作します．しかし，はじめに述べたように「思考」を助ける道具として使うことを考えると，X Windowが動作する環境が必要と考え，その環境を作ることを前提にします．ただ，ノート型パソコンなどのように制限の多い機種に導入する場合は，コンソール(DOSと同じような画面)で動作させ，仮想コンソールを使ってマルチタスクの環境を実現させます．

| TDE 500M | | | | | | | |
|---|---|---|---|---|---|---|---|
| /dev/hdal | * | 1 | 1 | 508 | 256000+ | 6 | DOS 16-bit>=32M |
| /dev/hda2 | | 509 | 509 | 559 | 25704 | 83 | Linux native |
| /dev/hda3 | | 560 | 560 | 625 | 33264 | 82 | Linux swap ← スワップ |
| /dev/hda4 | | 626 | 626 | 1017 | 197568 | 83 | Linux native ← ルート＋ユーザ |
| **SCSI 1000M** | | | | | | | |
| /dev/sda1 | * | 1 | 1 | 101 | 103408 | 6 | DOS 16-bit>=32M |
| /dev/sda2 | | 102 | 102 | 352 | 257024 | 83 | Linux native |
| /dev/sda3 | | 353 | 353 | 453 | 103424 | 83 | Linux native ← ホーム |
| (以下略) | | | | | | | |

〈表1-1〉
IDE 500 M バイト＋
SCSI 1000 M バイトの
ハードディスク

CPU には386以上が必要です．しかし，現状では486以上のCPUを搭載した機種が安くまた数多く出回っており，ほとんど問題はないでしょう．メモリは8Mバイト以上，できれば16Mバイト，ハードディスクは160Mバイト以上の空きが必要です．

ビデオカードは，640×480の白黒ならどのカードでも表示できます．チップとしてS3やET4000を用いたものなら，たいてい256色1024×768で使用可能です．PCIバスのカードにも対応しています．最新のビデオカードは避けて一世代前のカードを選べばよいでしょう．

XFree86 3.1.1(X Window)が対応しているビデオカードは大幅に増えましたし，Mach64にも対応しました．

SCSIカードはアダプテック社のAHA1542やAHA1522が無難でしょう．CD-ROMはSCSIはもちろん，ミツミやサウンドブラスターに付属のもの，ソニーなどに対応しています．必需品といってよいでしょう．

コンピュータを新たに購入するなら，各部品は最新のものではなく，すこし前にリリースされた製品がコスト的にもよいでしょう．たとえば，ビデオカードは少し前のカードにし，その分メモリとハードディスクへ投資したほうが快適な環境が作れます．

とくに，メモリは16Mバイトにすることをお薦めします．8Mバイトのメインメモリですと，ときどきスワップをし，その分動作が遅くなります．16Mバイトにするとスワップがたいへん少なくなります．

## 2．インストール

### 2-1．はじめに

はじめに，これから行うインストール手順をまとめます．

(1) Linuxをインストールするハードディスクの設計
(2) Linux立ち上げ用，その他のフロッピを3(または5)枚用意
(3) DOS上でCD-ROMからインストール用ディスクの作成
(4) インストーラを立ち上げて**fdisk**によるパーティション区切り(一番注意が必要なところ)
(5) 再立ち上げの後，**setup**によるインストール(ハードディスクのフォーマットその他の操作を含む)

これらがすんだら，

(6) X Windowの設定
(7) 日本語環境のインストール

### 2-2．ハードディスクの設計

Linuxをインストールする前に，ハードディスクのパーティション(区画)をどう区切るかを決めなければなりません．Linuxの場合もほかのOSと同様に，最低一つのパーティションが必要です．

一つのパーティションだけの場合，ルート(/)，バー(/var)，ユーザー(/usr)，ホーム(/home，自分の作業領域)，スワップのすべてを置くことになります．しかし，スワップは速度などの点から別のパーティションに分けたほうがいいでしょう．それで，ここではLinuxに最低二つのパーティションを割り当てることを考えます．

以下の区切り方は，ハードディスクの容量やPCの使用形態，さらには個人の考え方によりいろいろな場合がありえます．ここでは，**表1-1**のように区切りました．DOS，MS-Windowsと共存し，両方使えるようにします．

わかりにくいのですが，ここでは二つのLinuxを入れています．普段の仕事用と今回の連載用です．今回紹介するのは**/dev/hda4**をルート＋ユーザー，**/dev/hda3**をスワップ用，**/dev/sda3**をホームにしました．

みなさんのハードディスクの容量にもよりますが，ホームは，はじめから別のパーティションに分けておいたほうが良いと考えます．システム関係はCD-ROMからインストールをし直せばよいのですが，自分の作った文章などはバックアップをとっておかないと絶対に復元できません．

ルートとその他を一つのパーティションにとるということには，Linux のことをよく知っておられる方には「安全性」の立場から異議があるかもしれません．しかし，はじめて Linux を導入する場合，ほんとに使えるだろうかという気持ちがあり，とてもハードディスクから大きな容量を割けないでしょう．

したがって，はじめはとにかく使えるようになることを優先します．使えるという自信が出てきたら，ハードディスクを増設し，/, **/var**, **/usr**, **/home** を別のパーティションに分けましょう．

☞コメント

なぜ，こんなことをいうかというと，筆者が初めて Linux を導入したとき(SLS)，ルートに 10 M バイトをとりました．インストールの途中でエラーがでて，どうしてもインストールが先に進みません．かなり後でルートの容量が 2 M バイトほど足りないためとわかりましたが，そのために，2 週間ほど無駄にしました．

それよりも，悩んで悩んで Linux の導入をあきらめるところでした．ハードディスクの容量が十分になく，ルートに多少の余裕を持たせることをけちったためですが….

---

パーティションの設計ではありませんが，忘れないうちに，DOS の領域を DOS の **FDISK** を用いて確保します．

**2-3．Linux 立ち上げ用とその他のフロッピを用意**

フォーマットずみのフロッピディスクを 3 枚用意してください．使用する CD-ROM ドライブが MITSUMI, NEC-260, Sony の CDU535/351, CDU31A やサウンドブラスターに付属しているものの場合は，5 枚必要です．

**2-4．インストール用ディスクの作成**

CD-ROM から，最低限必要なインストール用プログラムをコピーします．

ここでは，InfoMagic 社の CD-ROM を例として説明します．CD-ROM の **DOS_UTIL** から **RAWRITE.EXE** と **GZIP.EXE** を，DOS のたとえば **C：¥LINUX** にコピーします．さらにブートディスクをディレクトリ(DOS では **DISTRIBU¥SLAKINST¥BOOT144**, Unix では **distributions¥slakinst¥boot144**)から選択してコピーします．

ブートディスクは，Linux をインストールするハードにより適切なものを選びます．筆者の CD-ROM は SCSI で，ネットワークカードがさしてあるので，

**SCSINET**

を選びました．ミツミの CD-ROM なら，

**MITSUMI**

サウンドブラスター用の CD-ROM なら，

**SBPCD**

を選びます．もしわからなければ，同じディレクトリにある **WHICH.ONE** を読んでください．

ルートディスクは，ほとんどの場合，ディレクトリ(DOS では **DISTRIBU¥SLAKINST¥ROOT144**, Unix では **distributions¥slakinst¥root144**)の，

**COLOR144**

を選べばよいと思います．もしインストールがうまくいかないようならば，ほかのファイルを選択します．

ここで，Linux の **slackware** というディレクトリがどこにあるか確認をして，メモしておきましょう．InfoMagic Linux の場合は，

**distributions¥slackware**

でした．この場合，Unix で用いる名前である必要があります．

用意したインストール用ディスクのそれぞれに，

1：INSTALL BOOT DISK
2：ROOT DISK
3：LINUX BOOT DISK

とラベルを貼っておきましょう．4，5 枚目が必要な構成なら，

4：MITSUMI または SB または NEC260 など
5：boot kernel

とします．この場合は MITSUMI 用またはサウンドブラスター用 CD-ROM をアクセスできるカーネルを，MITSUMI の場合は，

**DISTRIBU¥SLAKINST¥KERNELS¥S_MITSUM¥ZIMAGE，distributions/slakinst/kernels/s-mitsumi/zImage**

サウンドブラスターの場合は，

**DISTRIBU¥SLAKINST¥KERNELS¥S_SBPCD¥ZIMAGE，distributions/slakinst/kernels/s-sbpcd/zImage**

の **zImage** がカーネルになりますが，用意したフロッピにコピーしておきます．

もし，ファイルが圧縮されているなら，解凍しておきます(InfoMagic Linux の場合は圧縮されていない)．

▶コピーしたファイルの解凍：

```
C:¥LINUX> GZIP -d SCSINET.GZ
                        ← 選択したブートディスクのファイル名
C:¥LINUX> GZIP -d COLOR144.GZ
                        ← 選択したルートディスクのファイル名
```

▶インストール用ブートディスクの作成

フォーマットずみのディスクを A ドライブに入れて，

```
C:¥LINUX> RAWRITE
Enter source file name: SCSINET ← SCSINET を入力
Enter destination drive: A      ← A を入力
```

〈表1-2〉ハードディスクのデバイス名

| ディスク | デバイス名 |
|---|---|
| IDE 1台目のハードディスク | /dev/hda |
| IDE 2台目のハードディスク | /dev/hdb |
| SCSI 1台目のハードディスク | /dev/sda |
| SCSI 2台目のハードディスク | /dev/sdb |

しばらく待つとインストール用ディスク1ができます.

▶ルートディスクの作成

前と同様に，フォーマットずみのディスクをAドライブに入れて，同じ操作をします．

**C:¥LINUX> RAWRITE**
**COLOR144　　A**

を入力します.これで,インストール用のディスク2ができます．

**2-5．インストーラを立ち上げてfdiskによるパーティションを切る**

インストール用ブートディスク1をAドライブに入れ，マシンをリセットしてください．

**boot:**

というメッセージがでます．ここではリターンキーを押しましょう．

画面に目を凝らしてみていると，CD-ROMやハードディスク(IDEは**hda, hdb**, SCSIは**sda, sdb**と表示される)およびその他の周辺機器が認識されたというメッセージがでます．メッセージが画面の上に消えてしまったら，Shift＋PageUpとShift＋PageDownで画面を上下にスクロールすることができます．

もし,ここで,自分の使っているCD-ROMが認識されていなければ，**2-4**に戻ってブートディスクの選定をやり直してください．

しばらくすると，

**Please remove the boot kernel disk rom your floppy drive, insert a root/install disk(...**
**ramdisk and boot, and then press ENTER to continue.**

というメッセージが出ます．これはroot diskに入れ換えろということですから，ルートディスク2に入れ換えてリターンキーを押しましょう．

〈表1-3〉fdiskの使い方

| | |
|---|---|
| m | ヘルプ表示 |
| p | パーティション情報の表示 |
| n | パーティションの作成(追加) |
| t | パーティションのID変更.Linuxを入れるところのIDは83，スワップは82にする |
| d | パーティションの削除 |
| w | パーティション情報を書き込んで終了 |
| q | パーティション情報を書き込まずに終了 |

いろいろなメッセージがでて，

**slackware login:**

がでます．ここで，

**root**

と入力します．これは**root**という名前で**login**する操作です．いよいよPC-Unixの世界です．

DOSやMS-Windowsの世界からPC-Unixの世界に入るとき気をつけることは，入力する文字の大文字と小文字が区別されることです．Unixでは通常小文字を用います．それから，いきなり電源を切ることはできません．

先ほど「設計」をしたパーティションを実際に切ります．この操作を間違えるとDOSの領域を壊す可能性もあります．慎重に行ってください．

パーティションを切るために，**fdisk**を起動します．

**# fdisk /dev/hda**

なお,**/dev/hda**はIDEの1台目のハードディスクを示します．もし，Linuxをインストールするハードディスクが IDEの2台目ならば**fdisk /dev/hdb**とします．

**fdisk**やその他で使うLinuxにおけるハードディスクのデバイス名を**表1-2**に示しました．**fdisk**の使い方を簡単に**表1-3**に示します．

さきほど，操作は慎重にといったのは**fdisk**の起動のときに1台目のIDEハードディスク以外にLinuxをインストールするときです．うっかり**fdisk**とだけ入力すると，起動した**fdisk**は1台目のIDEのハードディスクを区切ってしまいます.**fdisk**を起動するときは必ず，ハードディスク名を付けてください．

**fdisk**が立ち上がったら,**p**を押すと現在のパーティ

```
# fdisk /dev/hda
Command (m for help) : p
Disk /dev/hda: 16 heads, 63 sectors, 1017 cylinders
Units = cylinders of 1008 * 512 bytes

Device      boot  Begin  Start  End   Blocks   Id  System
/dev/hda1   *         1      1  508   256000+   6  DOS 16-bit >=32M
/dev/hda2           509    509  559    25704   83  Linux native
```

〈図1-1〉パーティションの情報を表示

〈図1-2〉 パーティションの作成

```
Command (m for help) : n
Command action
   e   extended
   p   primary partion (1-4)
p
Partion number (1-4) : 3
First cylinder (560-1017) : 560 ←(はじめの数値を入れる)
Last cylinder or +sizeM or +sizeK (560-1017) : +32M
```

ション情報が表示されます．今はDOSや仕事用のLinuxのパーティションで275 Mバイト確保してあるので，図1-1には，**/dev/hda1**が250 MバイトでDOS領域，**/dev/hda2**がLinux用であることが表示されています．

**n**を入れてパーティションを作ります(図1-2)．

これで，スワップ用のパーティション32 Mバイトを確保しました．**n**を押すとパーティションが**extended**か**primary**か聞いてきます．

パーティション(区画)は1台のハードディスクに四つまでしか作れません．それ以上の区画を作りたいときは**extended**を選んで，**extended**区画を作ってからその中で論理区画を作ります．ここでは四つのパーティションを作る計画ですから，**extended**区画は必要ありません．

続いて，ルートその他を入れる区画を作ります．キーボードから入力する記号だけを書きます．

**n**(区画を作る) **p**(primary 区画)

**3**(区画ナンバー)

**626**(First cylinder) **1017**(endの値)

パーティションを切り終わったら，一度**p**で状態を表示してください．表1-1の前半のようになったはずです．

もし，決めたとおりにパーティションが切れなかったならば，修正するパーティション以降を削除して，作り直す必要があります．**d**を押して，削除するパーティションナンバーを指定します．

ほぼ予定どおりであれば，パーティションの**Id**を変更します．今の場合は，3番目のパーティション**/dev/hda3**の**Id**をスワップ用の**82**に変えます．**Id**の詳し

〈図1-4〉 installerの画面

| | | |
|---|---|---|
| HELP | Read the Slackware Setup HELP file | |
| KEYMAP | Remap your keyboard | |
| QUICK | Choose QUICK or VERBOSE install mode [now: VERBOSE] | ←① |
| MAKE TAGS | Experts may customize tagfiles to preselect packages | |
| ADDSWAP | Set up your swap partition(s) | ←② |
| TARGET | Set up your target partition(s) | ←③ |
| SOURCE | Select source media | ←④ |
| DISK SETS | Decide which disk sets you wish to install | ←⑤ |
| INSTALL | Install selected disk sets | ←⑥ |
| CONFIGURE | Reconfigure your Linux system | ←⑦ |
| EXIT | Exit Slackware Linux Setup | |

〈 OK 〉　〈Cancel〉

いことは，**L**を押すと一覧表が表示されます．**Id**の変更は**t**を用います(図1-3)．

これでよければ，**w**と押して保存します．**fdisk**は**w**を押すと記録して終了します．

つぎに，ホーム領域用の区画を作ります．同じ方法で作りますが，

**fdisk /dev/sda**

と入力します．デバイス名に気をつけてください．後は**/dev/hda**を区切った時と同じ手順です．

ここで区切ったパーティション情報を有効にするために，いったんLinuxを終了し，立ち上げ直します．Unixは一般的にいきなり電源offなりリセットをすることはできません(その必要もほとんどありませんが)．

さて，これで一番緊張する操作は終わりました．つぎの操作に移る前にコーヒーでも入れて飲まれるとよいでしょう．

では，boot用ディスクに入れ換えて，つぎの操作をします．

**# shutdown -r now**　　← 今(now)ということ

終了操作(**shutdown**)をして，再立ち上げ(**-r**)をしてしばらくすると，先ほどと同じ手順でLinuxが再起動します．

**2-6．いよいよインストール**

先ほどと同じ手順で再立ち上げをします．フォーマ

〈図1-3〉
Idの変更

```
Command (m for help) : t
Partion number (1-4) : 3
Hex code (type L to list codes) : 82
Changed system type of partition 3 to 82 (Linux swap)

/dev/hda3   560   560   625    33264   82  Linux swap
/dev/hda4   626   626   1017   197568  83  Linux native
```

〈表1-4〉インストールするソフトパッケージ

| | |
|---|---|
| **A** | Linuxの基本システム |
| **AP** | Xを必要としないアプリケーション |
| **D** | C，C++，Lisp，Perlなどやカーネルのソース(プログラミングに必要) |
| **N** | ネットワーク関係(TCP/IP，UUCP，Mail) |
| **X** | XFree86 3.1のX Windowシステム |
| **XAP** | X Windowのアプリケーション |
| **IV** | InterView関係 |
| その他にここではインストールしないが，つぎのソフトがある． | |
| **E** | GNU Emacs(日本語が通るemacsをJEから入れる) |
| **T** | TeX(日本語用のTeXをJEから入れる) |

〈表1-5〉インストーラお薦め(あらかじめX印がついている)以外に加えたソフト

A：
**tcsh**を加える．

AP：
**ispell jpeg workbone gp9600 groff sc**
そのほかに好みによってはシェルとして，
**ash zsh**
がある．

D：
**tools byacc man2 man3 extralib**
**f2c**……fortranからcのコードを作る．
**svgalib perl**

IV： そのまま．

N：
**ppp uucp smailcfg**

X：
**fvwmicns**
**Xserver：x_8514, x_agx, x_mach32, x_mach8, x_mono, x_p9000, x_s3, x_svga, x_vga16**
▶お薦め：自分のビデオカードのチップ用+**x_mono, x_vga16**

XAP：
**gs_x11，ghstview**を削る(後で日本語対応版を入れる)．
**gnuplot gchess xgames xpaint xspread**

XV：
**workman**

ットしてあるフロッピディスクを1枚用意してあるかを確認してください．立ち上げにより，画面には先ほどと同じ，

**slackware login:**

が表示されているはずです．

**root**で**login**して，

**# fdisk /dev/hda**

と，

**# fdisk /dev/sda**

を打ち込んで，先ほどのパーティションがきちんと区切られているか確認をしてください．**q**で**fdisk**を終了して，つぎに**setup**と入力して，インストーラを立ち上げます．

**# setup**

**図1-4**のような画面が出ます．大変良くできたインストーラです．いくつかのポイントを見ていきましょう．

はじめに**Q**を押してQuick modeにします(①)．続いて**A**を押してスワップのパーティションをセットします(②)．英語でいろいろとメッセージが出てきますが，スワップ領域を取るパーティションの入力は**/dev/hda3**と入れてください．あとは**Yes**を選択します．**Continue?**と聞いてくるので，**yes**を指定します．これで**Target**に入りました(③)．Linuxをインストールする**root**パーティションを聞いてきますので，ここでは先ほどの設計にしたがって**/dev/hda4**と入力します．

ユーザーの環境に合わせて**/dev/hda4**の部分を変えてください．

つぎに，ファイルのタイプを聞いてきます．標準の**ext2**にします．フォーマットするかどうかと聞いてきます．**Format，Check，Unformat**の三つが選択できます．はじめは**Check**を選択します．少し時間がかかりますが，ハードディスクをチェックしながらフォーマットしてくれます．

同じように**Continue?**と聞いてくるので，**yes**を指定します．これで**Source**に入りました(④)．Linuxが入っているメディアをセットします．ここでは**5**のCD-ROMを選択します．そして，CD-ROMの種類を選択します．ここでは**1 SCSI**を選びます．各自の環境に合わせて選択をしてください．

用いるdeviceは**/dev/scd0**にします．つぎに，Linuxが入っているディレクトリを選択します．ここではInfoMagic Linuxを使っていますから，それを選択します．それ以外の場合は**C**を選んでLinuxが入っているディレクトリを入力してください．

つづいて**Continue?**と聞いてくるので，**yes**を指定します．これで**Disk sets**に入りました(⑤)．インストールするシリーズを選択します．インストールしたい所に矢印で移動してスペースキーで**X**の印を付けます．ここでは**表1-4**のソフトパッケージを選択しました．

本当は**F**のFAQ listsを入れる必要があります．しかし，これはあとで追加インストールの練習として使う予定ですので，ここでは選びませんでした．

同じように**Continue?**と聞いてくるので，**yes**を指定します．これで**Install**に入りました(⑥)．

ここで問題になるのは，どのソフトをインストールしたらよいかわからないことです．ディスク容量に余裕があれば全部を入れるという手もありますが，普通はなけなしのディスクを割いて，Linuxを入れられる方が多いでしょう．筆者もはっきりとした基準はわか

〈表1-6〉
筆者がインストールしたソフトのリスト

disk a :

base devs idekern scsikern sysvinit ldso getty gzip ps shlibs util less bash comms cpio e2fsbn find grep keytbls lpr gpm shellutl syslogd tar textutl zoneinfo etc hdsetup lilo tcsh loadlin umsprogs

disk ap :

man joe diff bc gp9600 groff sc manpgs shlbsvga ispell

disk d :

tools gcc gxx flex include man2 pmake bison man3 libgxx libc svgalib m4 lx1159_1 lx1159_2 lx1159_3

disk f1 :

manyfaqs

disk iv1 :

iv_bin libiv_so

disk n :

netcfg tipip sendmail smailcfg uucp ppp

disk x :

x_s3 x_cfg fvwmicns xf_bin xinclude xlock xpm oldlibs xfnt xfnt75 config86 xf_lib xman

disk xap :

gnuplot libgr seyon vgaset xfileman xv gchess xgames xspread xpaint

りません．

しかし，ここではとにかく筆者の入れたソフトを書いておきます(**表1-5**，**表1-6**)．これはインストーラがお薦めのソフト(**X**の印が付いている)にいくつかを加えただけです．これがベストではありませんし，先に書いたように素人が故の無駄があるかもしれません．

もし足らなければ，あとで追加インストールすれば良いと考えます．

なお，Linuxがまだ対応していないCD-ROMを持っている方や，ブート用ディスクの選択を間違えた方は，ここでエラーが出て先には進めません．ブートディスクで立ち上げた時にCD-ROMが認識されていなかったのでしょう．**2-4**に戻って，持っているCD-ROMに合ったブートディスクを作り直してください．

持っているCD-ROMにLinuxが対応していない場合は，CD-ROMのSlackwareのディレクトリにある各ディレクトリの内容をハードディスクにコピーして，4番目のSourceの選択でハードディスクを選択します．

**表1-4**で挙げたパッケージをインストールするために必要なDOS領域のハードディスクの容量は，

disk a：3.8 M；ap：6.0 M；d：8.8 M；
iv：1.34 M；n：4.8 M；x：16.6 M；
xap：2.9 M；xv：3.0 M

合計で48 Mバイトぐらいです．x関係を除けば24 Mバイト程度の容量でいけます．

あとから，日本語環境JEを入れますので，ghostscript関係は「お薦め」であっても入れていません．また，emacs，texのパッケージも入れません．

**Continue?**と聞いてくるので，**yes**を指定し，**Configure**に入ります(⑦)．ここからは，あとでいくらでも変更ができます．しかし，初めての人にとっては質問ばかりで戸惑うところです．一つ一つ例を書いていきます．

最初のほうでbootdiskを作るかどうか聞いてきます．ここで**y**を入れ，はじめに述べたフロッピディスク3を入れます．network関係は**Hostname**と**domain name**だけ設定します．ここで設定した**Hostname**は，日本語を入力する設定で用います．

**CONFIGURE YOUR SYSTEMS?** **Yes**
**MAKE BOOT DISK?** **Yes**

ここで用意したフォーマットずみのフロッピを入れます．もちろん，書き込み可能にしておいてください．

**BOOT DISK CREATION** **Yes**

フロッピに書き込みをします．

**MODEM CONFIGURATION** **Yes**
**SELECT CALLOUT DEVICE**
**cua1 com2 : under DOS**を選ぶ． **OK**
**MOUSE CONFIGURATION** **Yes**
**SELECT MOUSE TYPE**
**Mouse System serial mouse**
(3ボタン式はほとんどがこのタイプ)なら **6**
**Microsoft serial mouse**なら **1**
**PS/2**マウスなら **2** を選び， **OK**
**SELECT SERIAL PORT**
**ttyS0 com1 : under DOS**を選び， **OK**
**SCREEN FONT CONFIGURATION** **No**

もしコンソール画面のフォントが気に入らなければ，ここでセットします．

**SET YOUR MODEM SPEED 38400** **OK**

モデムの最高スピードを選ぶから**38400**にセット．

**LILO INSTALLATION**
**Skip**を選ぶ．

もし，ハードディスクからLinuxを起動をしたいな

ら，LILO の設定をします．設定の詳しいことは後ほど述べます．

**CONFIGURE NETWORK?** **Yes**

ネットワークにつなぐ予定がなくてもこの設定は必要．

**NETWORK CONFIGURATION** **OK**

**ENTER HOSTNAME** **sato**

自分のマシンに名前を付けてやってください．ここでは筆者の名前の一部を取り出して付けました．

**ENTER DOMAINNAME** ***.**.jp

ドメイン名の指定です．ネットワークにつながっていない場合は何でもかまわないのですが，系統的に決める方法としては自分の住所から作ります(詳しくはトラ技コンピュータ 1994 年 12 月号 Linux マシンとパソコンを接続する方法 p.90 参照)．

名古屋市中村区なら，

**someone.nakamura.nagoya.jp**

愛知県蒲郡市なら

**someone.gamagori.aichi.jp**

とします．

**LOOPBACK ONLY** **Yes**

ネットワークにつなぐ場合は，ここで **No** を選択し設定をします．しかし，これは必要になってから行います．つぎに進みましょう．

**GPM CONFIGURATION** **Yes**

コンソールでマウスが使えるようにセット

**SENDMAIL CONFIGURATION** **SMTP OK**

**TIMEZONE CONFIGURATION** **Japan OK**

**SETUP COMPLETE** **OK**

これで，はじめの **Setup** 画面に戻りました．ここで，

**EXIT** **OK**

DOS で見慣れたコンソールの画面に戻りました．

**#**

が表示されているはずです．これでインストールは終了です．

SCSI の CD-ROM を使われている方はこのまま終了操作をして，再起動すればあなたはめでたく PC-Unix のオーナーです．

```
# sync ; sync
# shutdown -r now
```

で再起動します．または電源を切る時は **sync** をした後，

```
# shutdown -h now
```

として，しばらくすると，

**halt**

と表示されれば，電源を切ることができます．

これで，とりあえず Linux のインストールはできました．まだ，SCSI 以外の CD-ROM が使えないなどの不都合が残っていますが，コンソールでいろいろなコマンドを試すことはできるはずです．誌面が尽きました．Unix 関係の本を買い込んできて，あれこれとコマンドを実行させてみてください．

次回は CD-ROM の設定と Unix の雰囲気を味わうことをした後，X Window の設定をする予定です．

# REXX言語によるOS/2入門

## 4 VisPro/REXX(最終回)

山下直彦

連載の最終回の今回も，REXXよりビジュアルな環境で開発ができるVisPro/REXX(注1)について述べます．また，VX-REXXとの比較についても触れたいと思います．

### VisPro/REXXによる例題(1)

VisPro/REXXを使った最初の例題プログラムをディレクトリ**C:¥VPR1**に収納するものとします．VisPro/REXXのアイコンを開きます．すると，ほかのアイコンに混ざってProjectsのアイコンが表示されます．これをダブルクリックすると**VPR Database.dbd**と**PROJECT**のテンプレートが現れます．

後者をマウスボタン2でドラッグしてC:ドライブアイコンにドロップします．そして，フォルダ名を**VXR1**に変更します(ALTキーを押しながら，フォルダ名をマウスボタン1でクリックすると現れる編集フィールドで，名前の変更が可能)．このフォルダを開くと，**Threads**，**SubProcs**，**Main**および**Form**アイコンが現れます．

● **画面設計**

この**Main**を開くと，**図4-1**に示した設計環境が現れます．VisualBasicあるいはVX-REXXと似た点が多々出てきますが，相違点もかなりあります．

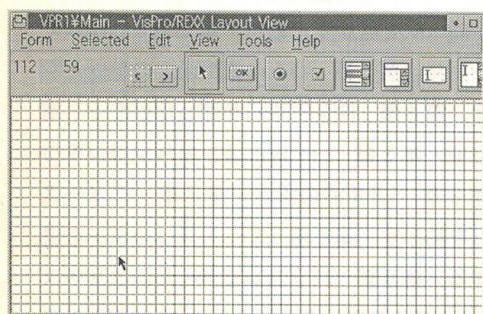
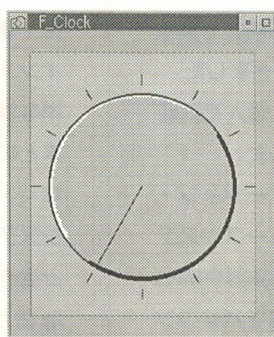
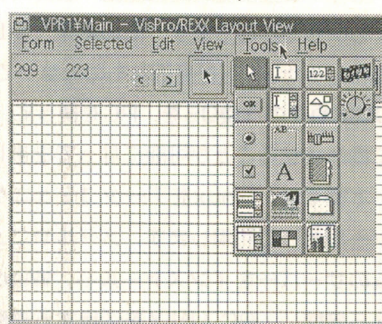
また，この例題の完成時の姿を**図4-2**に示します．問題の設定としては，起動すると，時計に模した円盤が現れ，針が12時の位置から1秒ごとに1時間分ずつ進むものとします．

メニューバーの下にあるのは，Toolバーです．1行に納まらないので，矢印ボタンをクリックしてToolを順次見ることができます．あるいは，メニューからToolsを選ぶと，**図4-3**に示した**Tools palette**が現れ全体を見渡すことができます．また，後ほど述べる方法でToolバーを表示しないようにもできます．

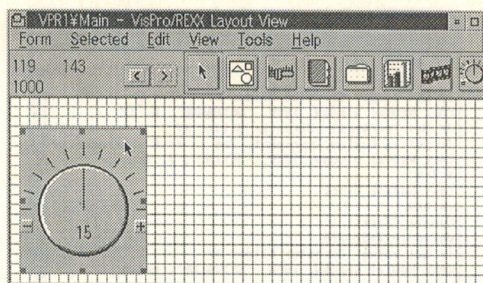
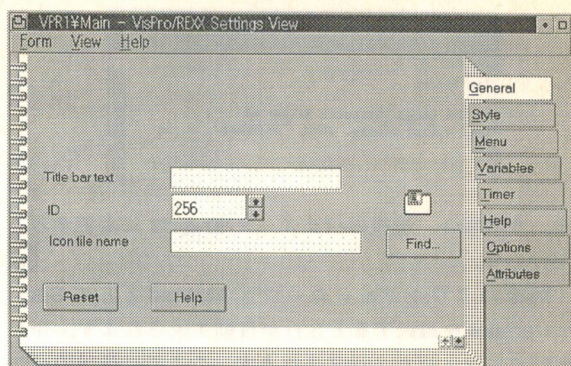
Toolバーの下の破線で方眼が引かれた領域は**Canvas**と呼ばれ，ここに，**Object**を配置して画面の設計を行います．また，設計環境の画面全体を**Form**(今の場合は**Main Form**)と呼んでいます．

**Circular Slider**(注2)をマウスボタン1で選び画面に張り付けます．マウスを**Canvas**に移動するとポインタの形が変形(対応するToolのアイコンが付加される)します．張り付けたい領域の左下の角の座標をマウスボタン1でクリックします．

マウスの位置をずらすとマウスポインタが変形したままです．その後，Toolバーの**Pointer**をクリックするとポインタの形がもとに戻ります．**図4-4**に示すよ

〈図4-1〉 VisPro/REXXの設計環境

〈図4-2〉 例題(1)で作る時計もどき

〈図4-3〉 TOOL palette

(注1) V 2.03 Gold Editionによる．

(注2) これとMMPM/2 videoはMMPM/2を事前にインストールしている時にのみ現れる．

〈図4-4〉 リサイズができる状態

〈図4-5〉 Setting View を開いたところ

うに，張り付けた **Object** には 8 個の黒い正方形のマークが表示されていますので，必要なら領域のサイズの変更ができます．そのためには，マウスポインタをこのマークに合わせボタン 2 でドラッグします(この点，ボタン 1，2 の使い分けが通常と異なる)．また，領域内をボタン 2 でドラッグすれば，移動することができます．

Tool バーの左端に数字が，上段に二つ表示されます．これはマウスポインタの x, y 座標を表します(**Canvas** の左下の角が原点 mm 単位)．グリッド間隔を，デフォルトの **25** から **10** に変更(後述)しておくと座標が見やすくなります．

マウスポインタが今張り付けた **Object** の領域に入ると，下段に数字(今の場合は **1000**)が現れます．これは **Object** の **ID** 番号で，生成の順に **1000** からの連番が自動的に生成されます(後述の方法で **Symbol** 名を付けると，それが **ID** 番号の代わりに表示される)．

この例題で，**Object** の張り付けは以上で終わりです．次に，**Main** の状態設定に移ります．この設定ノートブックの表示方法は，メニューから **Form** を選び，次に **Open** を選び，**Settings** を選ぶ手順で行います．慣れるまで，このウィンドウサイズは大きめに選びましょう(図 4-5 参照)．

General 頁の **Title bar text** に **F_Clock**(注3)を設定，また，Style 頁の **Plain** を **Sizable** に変更，Timer 頁の **Duration** を **10**(0.1 秒単位)つまり 1 秒に設定します．また，Timer 頁には，**When expired** というボタンがあります．これは後ほどふれます．最後に，Options 頁には重要な設定項目として，グリッド関連のサイズ指定，Tool バーを表示するかどうか，また，編集途中の自動セーブなどがあります．

次に，**Circular Slider** の状態設定に移ります．この場合は，**Object** をマウスボタン 1 でダブルクリックすると現れる点は，VX-REXX と同様です．Style 頁の **Exclude buttons**，**Exclude number** および **360 degrees** を選択，Attributes 頁の **Background Color** を **Green** および **Blue** をともに **255** にします．

● **コーディング**

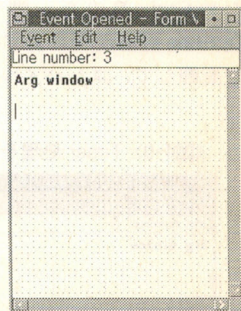
まず，Tool バーが不要となりますので非表示にします．その手順は，メニューから **Form**，**Open**，**Settings** で設定ノートブックを表示し，Options 頁から **Show tool bar** のチェックボックスを非選択とします．

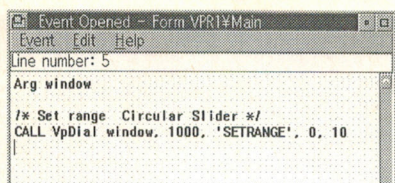
次に，プログラム起動時のコーディングに移ります．今度はメニューより，**Form**，**When** を選択すると **Event** の選択リストが現れるので，**Opened** を選択します．すると，**図4-6** のような **Event Opened-Form VPR1¥Main** と名付られた **Event** ウィンドウが現れますので，ここにコードを作成していきます．

最初に，**Arg window** という 1 行の REXX 文が前もって書かれています．このサブルーチンが呼ばれたとき，仮変数 **window** には **Main Form** に対する内部コードが渡されます．

まず，**Slider** の初期設定をしましょう．**Form** の **Circular Slider** をマウスボタン 2 で **Event** ウィンドウにドラッグ&ドロップします．すると，**Create Link** のウィンドウが現れます．リストボックスから **Method** を選択します．**Set range** を選択すると**図4-7** のように 2 行の REXX 文が生成されます．

最初の行はコメントで，2 行目はテンプレート文で

〈図4-6〉
**プログラム起動時の処理を記述する Event ウィンドウを開く**

(注 3) VX-REXX と同様に日本語の表示もできる．そのときは，Font の設定に気を付けることも同様である．

〈図4-7〉 Slider の記述

す．今の場合，最後のパラメータ **10** を **12** に変更するだけです．

同様のドラッグ&ドロップで **Method** として **Set increment** を選択すると3行が自動生成されます．今度はコメント行が2行含まれます．ここでは最後のパラメータを **5** から **1** に変更します．

次に経過時間(秒)を表す変数 **v_T** を導入し，

```
v_T=0
call Update_Clock
return
Update_Clock:
v_Csld=(v_T+6)//12
```

のようにコードを入力した後，**Circular Slider** をこのウィンドウにドラッグ&ドロップし **Set value** を選択します．なお，演算子//は整数演算の剰余を表します．

以上で，プログラム起動時の設定(図4-8)を終わりました．

残るコーディングは，**Timer** の設定時間(1秒)が経過したときの処理の記述です．そのため，メニューから，**Form**，**Open**，**Settings** で設定ノートブックを呼び出し，Timer 頁を開き，**When expired** のボタンをクリックします．すると，図4-9のような **Event** ウィンドウが現れます．

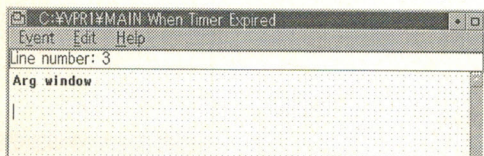
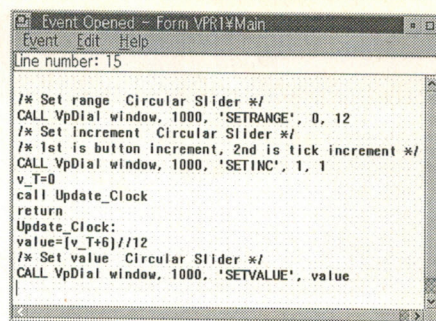
```
v_T=v_T+1
call Update_Clock
```

と入力するとコーディングは終わりです．

### ● 試行およびコンパイル

プログラムの試行は Ctrl+**R** で実行できます．プログラムの試行を行うと **_RUN.CMD** というファイルが生成されます．また，Ctrl+**T** でデバッグを行うこともできます．

プログラムが完成すれば，コンパイルします．そのため Ctrl+**B** のキー押すと，図4-10に示した **Build Options** 画面が現れますので設定をした後，**Build** ボタンを押して，コンパイルを実行します．**_RUN.CMD** が

〈図4-9〉 Timer 処理を記述する Event ウインドウ

〈図4-8〉 プログラム起動時の設定が終わった

消えて **RUN.EXE** が現れます．

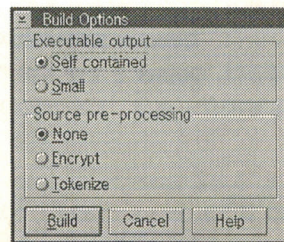
ここで，図4-10のオプションの簡単な説明をしておきます．まず，**Executable output** の意味ですが，実行時に **VPREXX.DLL**(**CONFIG.SYS** で **LIBPATH** に設定されたディレクトリに納める)を必要とするかどうかの選択です．なお，特に，**Circular Slider** あるいは **MMPM/2 video** を使用したときには，それぞれ **VPDIAL.DLL** あるいは **VPVIDEO.DLL** がさらに必要です．

また，**Source pre-processing** の選択で，REXX のソースコードはそれぞれ，

**None** ：そのままの形
**Encrypt** ：暗号化されて
**Tokenize**：トークン化されて〔連載第1回の(注3)参照〕

**EXE** ファイルに埋め込まれます．上記の説明でコンパイルとの言葉を使いましたが，ビジュアルなプログラム環境ではその手前で止められるのはよくあることです．

この簡単な例では，ソースは **Main** フォルダのみにあります．そのため，プログラミングの開始時，C：ドライブに **VPR1** のディレクトリを作り，このディレクトリに **Project** テンプレート内の **Main** フォルダを複写するという方法もとれます(このときは Ctrl キーを押しながら，ドラッグ&ドロップする必要がある)．ただし，次の例題のように，二次 **Form** も使用する場合には，最初に述べた方法のほうが簡単でしょう．

〈図4-10〉
ビルドのオプションを設定する画面

## VisPro/REXXによる例題(2)

次の例題は，C:¥VPR2に作成するものとします．ビットマップファイルの表示を行うプログラムです．

このプログラムを実行すると，図4-11のようなファイルダイアログが現れます．ここで，ファイルの指定を行うと，そのディレクトリ内のファイルアイコンとファイル名が並んだ図4-12のContainerが現れます．この中から一つのBMPファイルのアイコンをクリックすると，対応するグラフィック(この実行例ではBUTTERFL.BMP)が図4-13のように第2のFormに表示されるという仕様です．

先の例題と同様にして，C:にProjectを複写しVPR2と名前を変更します．今回は二次Formも使用します．VPR2のフォルダを開くと，ほかのアイコンに混じってFormテンプレートが現れます．これをこのフォルダ内のスペースに，ドラッグ&ドロップして複写すると，Form:1が生成されます．

Mainのアイコンをダブルクリックして，Toolバーから CavasにContainerを張り付けます．

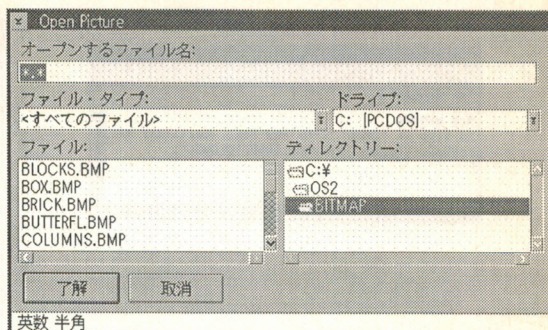
Main Formの設定としては，General頁のTitle bar textにBMP_Viewer，Style頁でSizeableを設定します．

Containerの設定としては，Style頁のSelectionをSingleに設定します．

Form:1のアイコンをダブルクリックして，Graphicを張り付けます．Form:1の設定としてはStyle頁にSizeableを設定します．以上で画面設計は終了です．

なお，先述の方法で作成した第2のFormは，自動的に次のように設定されます．

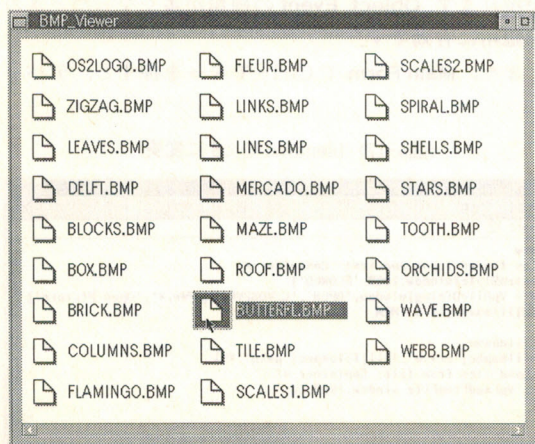
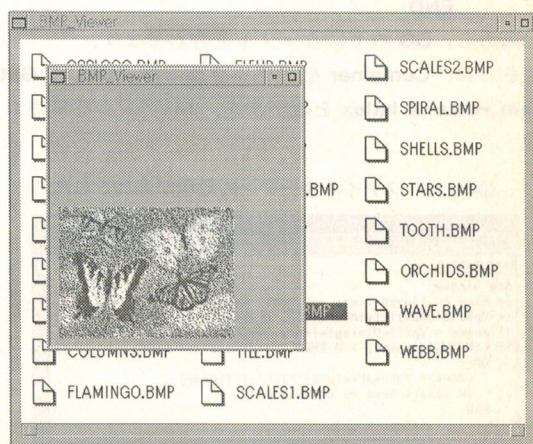
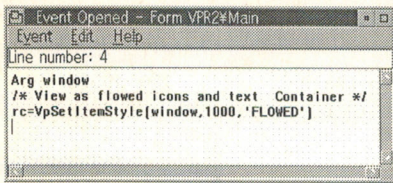
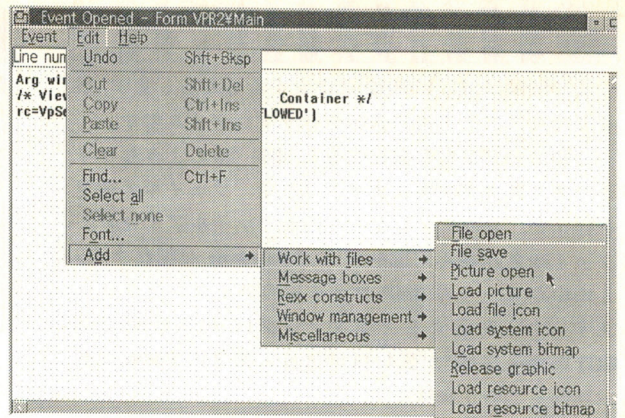
(1) メニューバーのFormのプルダウンメニューにCreate linkが追加される．

〈図4-11〉例題(2)で作成したプログラムの実行画面

(2) FormのGeneral頁に表示されるIDとして257以降が自動的に割り付けられる(厳密には，Form:1をダブルクリックして開発環境を呼び出したときに割り付けられる．したがって，二つ以上の二次Formを生成したときは，これらをダブルクリックして呼びだした順序で257，258，…と割り付けられる)．一方，Main Formのそれは256である．

(3) FormのStyleの頁にNotebook pageのRadioボタンが追加される．

次はコーディングの番です．まず，起動時の処理です．メニューからForm，When，Openedを選んでEventウィンドウを表示します．Containerからこのウィンドウにドラッグ&ドロップして，MethodからView as flowed icons and textを選択します(図4-14参照)．

次に，Eventウィンドウのメニューの Edit，Add，Work with files，Picture open(図4-15参照)を選びます．なんと，図4-16に示すように6行のテンプレート文が現れます．ただし，ここでは，やや変則的な使い方をしていますのでDO ENDで挟まれた2行を消します．そして，

〈図4-12〉BMPファイルのコンテナを表示

〈図4-13〉選択したBMPファイルが表示される

〈図4-14〉 Event ウィンドウに View as flowed icons and text の処理を追加

〈図4-15〉 Event ウィンドウにメニューを追加

**fn＝filename**

**cd＝filespec ( 'drive', fn)||filespec ( 'path', fn)**

と入力します．ここに **filespec( )** は REXX の組み込み関数で，ファイル名のフルパス名から **drive**，**path**，**name** などを抜き出すために使用できます．また，**VpFileDialog** 関数の第 3 パラメータを **'C:¥OS2¥BITMAP¥*.*'** に変更します．

さらに，**Container** からこのウィンドウにドラッグ&ドロップして，**Method** から **Load item from file** を選びます．最後のパラメータの **filename** を **cd** と変更します(**図4-17** 参照)．

次に，**Container** から項目をダブルクリックしたときのコーディングをしましょう．**Container Object** をマウスボタン 2 でクリックするとメニューが現れます．これから **When**，**Mouse button 1 double click** を選びます．このウィンドウにコーディングしていきます．

**Container** をドラッグ&ドロップして **Get index of first selected item** を選びます．次に，

```
IF index > 0 THEN
    DO

    END
```

と入力し，**DO** の下にカーソルを合わせます．

さらに，**Container** をドラッグ&ドロップして **Get item value at index** を選びます．2 行の文が生成され，その下にカーソルを合わせます．

次に，**Form : 1** をダブルクリックして，メニューから **Form**，**Create link** を選びます．この直後には顕著な変化は現れません．

先の **Event** ウィンドウに戻り Shift＋Ins キーを押します．2 行のテンプレート文が現れます．**'topic name'** を **cd||value** に変更します(**図4-18** 参照)．なお，**topic name** の項は，第 2 **Form**(この場合 **ID257**)の **Topic** という名前の変数に代入されます．

次に，**Form : 1** のコーディングをします．メニューから **Form**，**When**，**Opened** を選び，

**filename＝topic**

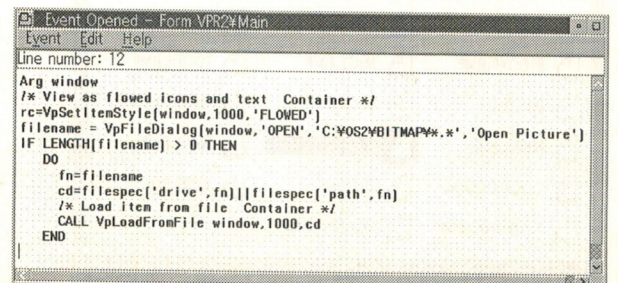
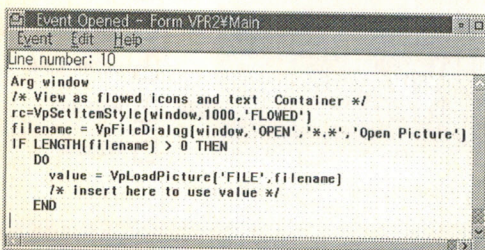
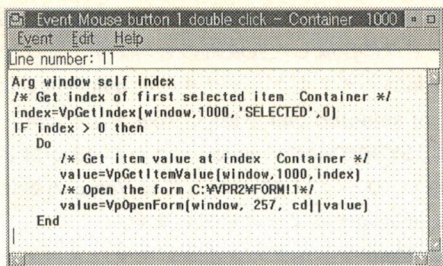
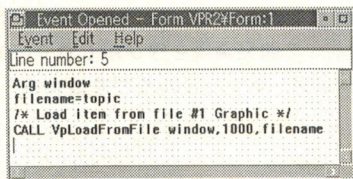
と入力します．

こんどは，**Graphic** をドラッグ&ドロップして **Load item from file** を選びます(**図4-19** 参照)．以上でコーディングは終了です．

なお，メニューから **View** を選ぶとプルダウンメニューが現れます．通常 **Layout view** にチェックマークが入っています．**Event tree view** を選択すると，**Form** の表現が**図4-20** のように変化します．さらに，マウスのアイコンをクリックすると，対応する **Event** の処理が現れます．**Object, Event** の種類が多くなったときの確認用に有効です．

また，**Main Form** で Ctrl＋**T** キーを押すと，デバッ

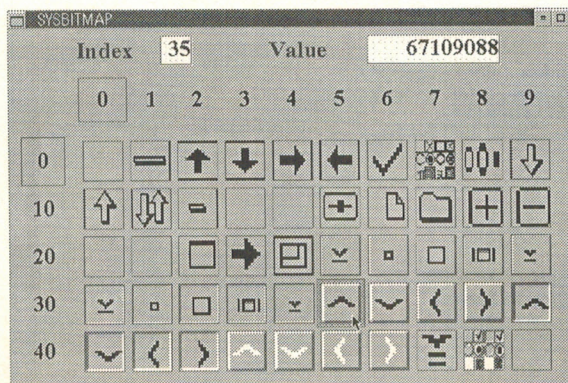
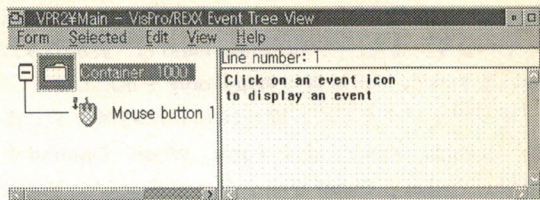
〈図4-16〉 ファイルオープンの処理が追加された

〈図4-17〉 filename を cd に変更

〈図4-18〉 topic name を cd||value に変更

〈図4-19〉 Load item from file の処理を追加

ガが起動され**図4-21**のような画面が現れます．さらに，**Event**欄の＋印をクリックするとトリガとなる**Event**がアイコン付きで説明されます．つぎに，このアイコンをダブルクリックすると，対応するルーチンのコードが現れます．また，**Form：1**の情報も一連の操作で見ることができて便利です．

なお，**_RUN.CMD**の抜粋を**リスト4-1**に示します．内部処理の一端を想像することができます．

## VisPro/REXX による例題(3)

次の例題は，ディレクトリ **C：¥VPR3** に作るものとします．**Project** テンプレートをC：ドライブに複写して，**VPR3** と名前を変更します．この例題の完成時の姿を**図4-22**に示します．5行10列の **Value Set** に **System Bitmap** を配置し，任意のセルをマウスボタン1でダブルクリックすると，対応する **Index** および **Value** を表示するものとします．

〈図4-22〉 例題(3)の実行画面

〈図4-20〉 Event tree view を活用

〈図4-21〉 デバッガを起動した様子

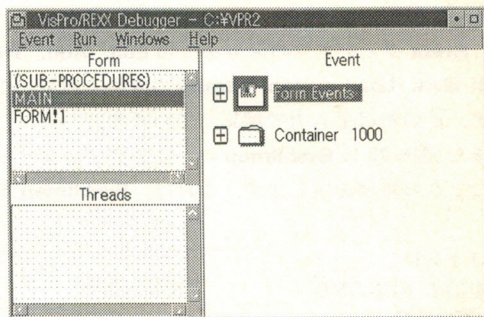
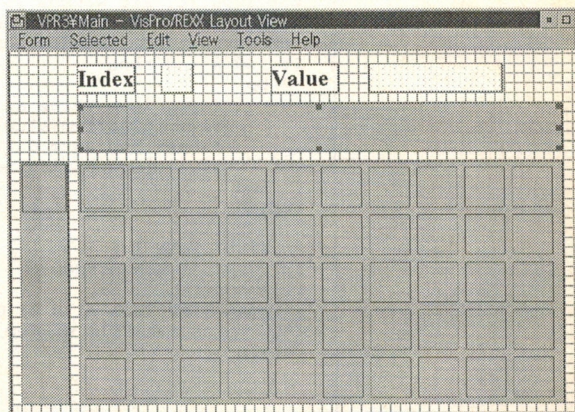



**図4-23**に示したのは画面設計を一通り終えた時点の画面です．**Value Set** の各セルのサイズは5mm四方としています．この例のように，規則的な画面の場合，**Object** の配置はビジュアルには大まかに行い，各々の設定ノートブックの **Position** で最終的にディジタルで位置とサイズを決めたほうがやりやすいでしょう．

**Main Form** の設定は，Genaral 頁の **Title bar text** に **SYSBITMAP**，Attributes 頁の **Foreground Color** の **Blue** に **255**，**Font** に **18** の **Tms Rmn Bold** を設定しています．

三つの **Value Set Object** を使っています．まず，**Index** の **10** の桁を表すために5行1列のもの，1の桁を表すために1行10列のものを当てています．最後に

〈図4-23〉 画面設計が終わったところ

**Sysbitmap** の表示用で，Style 頁の **Cell contents** は **Bitmap** また，**Scale bitmap** および **Item border** を選択しています．また，二つの **Entry Field** は，表示のみに使いますので，Style 頁の **Read-only** を選択します．

コーディングとして，図4-24 に初期処理を示します．これは，メニューから **Form**, **When**, **Opened** を選択して現れる **Event** ウィンドウに作成した完成時のものです．三つの **Value Set** の初期化のために，各々からこの **Event** ウィンドウへドラッグ&ドロップし **Method** として **Set cell value** を使用しています．また，**Event** ウィンドウのメニューの **Edit**, **Add**, **Work with files**, **Load system bitmap** の機能を利用しています．その他は手入力で編集しています．

また，図4-25 に **Sysbitmap** の一つをダブルクリックしたときの処理を示します．これは，**Sysbitmap** 表示用の **Value Set** をマウスボタン２でクリック，**When**, **Mouse button 1 double click** を選択して現れる **Event** ウィンドウに作成した完成時のものです．ここでの最初の作業は，ダブルクリックされた **cell** の同定に **Get index of selected cell** を使い，**row** および **col** を得ています．**SYSBITMAP** の **index** は **1** から **48** が許容されますので，**0** および **49** に対しては **Value** の欄に **'No Valid!'** を表示するようにしています．

なお，**Main** フォルダのアイコンをマウスボタン２でクリックすると，図4-26 に示したようなプルアップメニューが現れ，図のようにさらにオープンを選び，**Main Form** の **Settings** ノートブックを呼び出したり，**View** を指定して開発環境を呼び出したり，あるいは，実行，デバッグ，コンパイルなどを直接行うこともできます．

〈リスト4-1〉
例題(2)の_RUN.CMD
の内容の抜粋

```
/* VisPro/REXX */

_VPAppHandle = VpInit()
if _VPAppHandle='' then exit
_VPAppReturn = 0

/* Open first form */
Parse arg topic
_VPMainWindow = VpOpenForm('', 256, topic)
if _VPMainWindow='' then do
   _VPAppReturn = '-1'
   signal _VPAppExit
end

中略・・・

/* ##256_FORM.1 */
256_FORM.1: Arg window
/* View as flowed icons and text  Container */
rc=VpSetItemStyle(window,1000,'FLOWED')
filename =
VpFileDialog(window,'OPEN','C:\OS2\BITMAP\*.*','Open Picture')
IF LENGTH(filename) > 0 THEN
   DO
     fn=filename
     cd=filespec('drive',fn)||filespec('path',fn)
     /* Load item from file  Container */
     CALL VpLoadFromFile window,1000,cd
   END
return

/* ##256_1000.16 */
256_1000.16: Arg window self index
/* Get index of first selected item  Container */
index=VpGetIndex(window,1000,'SELECTED',0)
IF index > 0 then
   Do
      /* Get item value at index  Container */
      value=VpGetItemValue(window,1000,index)
      /* Open the form C:\VPR2\FORM!1*/
      value=VpOpenForm(window, 257, cd||value)
   End
return

/* ##257_FORM.1 */
257_FORM.1: Arg window
filename=topic
/* Load item from file #1 Graphic */
CALL VpLoadFromFile window,1000,filename
return
```

## VX-REXXとVisPro/REXXの比較

最後に，両開発環境(以降前者をVXR後者をVPRと略記する)を使用してみて気付いた,相違点の顕著なものについて述べたいと思います〔比較に使用したバージョンは前者がV 2.1 aC/S，後者は(注1)を参照，いずれも英語版〕.

(1) マスタディスケットの枚数：VXRが5枚(Standard版は3枚)に対し,VPRは1枚(いずれも2HDベース)である．両方とも英語版だが，MS-Windowsのソフトに比較するとコンパクトである．OS/2システムの機能を上手に利用しているためであろう．

(2) ノートブックでの設定：VXRは詳細で**Object**により異なる要素が多い．VPRは簡略で**Object**による相違はStyle頁に集中，また**Method**関数でのみ設定できる項目も多い．

(3) 実行時のウィンドウサイズ：VXRは設計時に依存，VPRは設計時の**Object**の配置から適当なマージンをシステムが設定して表示するという方式が取られる．

また，**Sizeable**を選択したとき，実行時にウィンドウサイズを変えると，各々の**Object**のサイズも連動して変わる．

(4) **Object**の複写削除など：VXRは対応する**Event**ルーチンも複写などがされるが,VPRではルーチンは生成や削除が行われない．

(5) REXX文の自動生成機能：VPRには，**Event**ウィンドウのメニューの**Edit**，**Add**機能を利用することにより，大幅な自動生成が可能である．また，注釈文が付記されるので，パラメータの意味などがつかみやすい．

(6) ヘルプキーF1：VXRのほうがヘルプは詳細，時々の操作に対応した，的確なヘルプが現れる．

(7) **Object**の種類：VXRにのみあるものは“**DDE Client**”，一方VPRのみにあるものは，“**Circular Slider**”，“**MMPM/2 video**”である．ただしデータベースへのアクセスの扱いは，両者で異なるので，その関連の**Object**には，ここでは触れない．

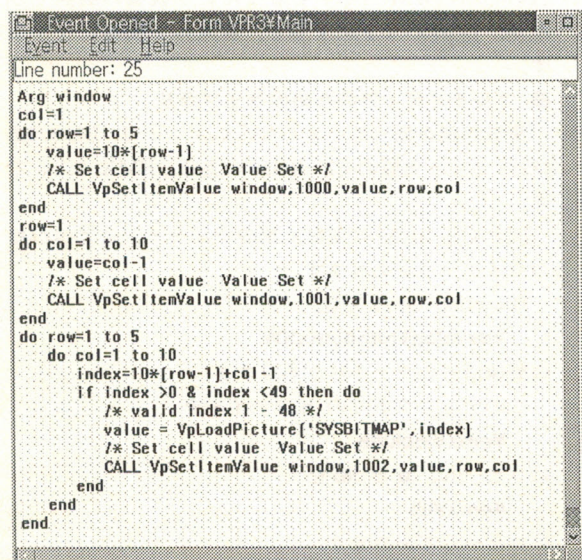
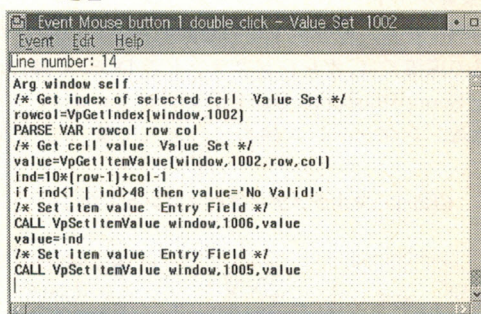
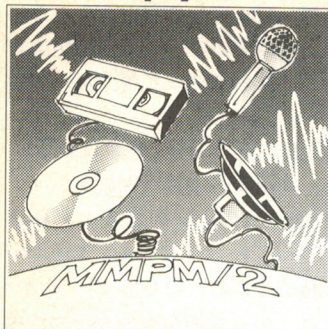
(8) **EXE**ファイル形式：実行時ライブラリのサイズはVXRが約875 Kバイト,一方VPRは約158 Kバイトである．このこともあり，VXRの**EXE**ファイルは実行時ライブラリが別に必要な一形式(Small，Tokenize相等)であるが，VPRの場合は実行時ライブラリを必要としない形式などの多くのケースが扱える．

(9) その他：VPRは設計環境に**Layout**，**Event tree**，**List**のいずれかの**View**が選択できる．また，デバッグモードもビジュアルでわかりやすい．

連載を終わるにあたり，説明が不十分でわかりずらかい点もあったと思いますが，できれば別の機会に補足したいと考えます．

◆ 参考文献 ◆


(1) HockWare；VisPro/REXX's Programming Manual.


〈図4-24〉初期処理のコーディング部分

〈図4-25〉Sysbitmapの一つをダブルクリックしたときの処理

〈図4-26〉アイコンをダブルクリックして出てくるプルアップメニュー

# GNU C/C++ Compilerでマスターする OS/2プログラミング

## ⑥ 初めてのMMPM/2プログラミング

岡田好一

### ● OS/2でマルチメディア！

筆者がはじめてマルチメディアに触れたのは，Sunというワークステーションで，3年ほど前のことす．

マイクとCD-ROMドライブが付いていて，すてきなCD-ROMが何枚もおまけで付いていました．もちろん，すぐに「これからは，これだ！」と思いました．

しかし，ワークステーションでマルチメディアをプログラミングする方法は容易にわからず，ユーティリティもほとんどなし，という状態ではどうにもならず，結局Windows 3.1を待つことになりました(**図6-1**)．

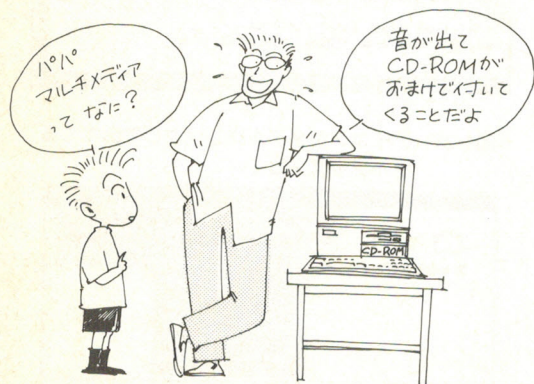
OS/2のマルチメディアはMMPM/2(Multimedia Presentation Manager/2)と呼ばれ，プログラマが気軽に利用できます．32ビットとマルチタスクの威力はなかなかのもので，気楽にプログラミングできる，という点においては，最高の環境のひとつと言ってよいでしょう．

## MMPM/2のために用意するもの

### ● 音源ボード

当然ながら，オーディオのための音源が必要です．筆者の環境では，ガレージブランドのDOS/V機に，有名なSoundBlaster16 SCSI-IIを付けています．

最近，レファレンス機の一つとして使っている，

〈図6-1〉 マルチメディアパソコンってなに…

Compaq Presario520 CDSには，音源が内蔵されています．

Presario520では，MMPM/2のインストール時に，若干の設定(IRQ＝5)が必要でした．SoundBlasterはうまく認識でき，インストール時に設定しなおす必要はありませんでした．

### ● FPUとメモリ

マルチメディアプログラミングには，浮動小数点機構(FPU)が必須と考えてください．整数だけでも何とかなりますが，プログラミングの労力ばかりかかって，計算速度はあまり変わりません．

最近では，オールインワンのパソコンにすら，FPU付きのCPUが採用されています．本稿では，FPUを前提に話を進めます．

筆者のPresarioは，最初は486SX2-66 MHzでOS/2も快適だったのですが，マルチメディアのプログラミングを始めるとがまんできなくなり，DX4-100 MHzに増強してしまいました．

メモリは12Mバイトに増やしています．初めから付いていた8Mバイトでもなんとか我慢の限界でしたが，12Mバイトにすると，目立って快適になります．

### ● マルチメディアのためのライブラリ

筆者が現在使っているGNU C/C++は，Blowfish OS/2という市販のCD-ROMにある，EMX 0.9aです．

同じCD-ROMにEMX用のマルチメディアのためのライブラリがあります．

Blowfish OS/2では，

**¥develop¥mm4emx10¥**

に解凍済みのファイルがあります．

このディレクトリの**.h**ファイルを

**¥emx¥include**

にコピーし，**.a**と**.lib**を

**¥emx¥lib**

にコピーしておきます．

コンパイル時に，マルチメディアのライブラリを指定します．例えば，

**gcc -o mmtest1.exe mmtest1.c mmtest1.def -los2me -Zomf -Zmtd -s**

などとします．**-los2me** がライブラリの指定です．

● MMPM/2 のためのドキュメント

唯一困るのは，ドキュメントです．BonusPak には見当たりません．

Toolkit(IBM C Set++にも付属している)を買うと **MMPROG.INF**(英語)が手に入ります．

一番よいのは，Warp の Developer Connection の CD-ROM にある，Warp のオンラインドキュメント・セット(英語版)です．開発者向きの資料ですが，誰でも手に入ります．

「マルチメディア」フォルダーのマルチメディア REXX(**図6-2**)と，mm4emx10 のサンプルプログラムからある程度までは類推できます．

しかし，C/C++による MMPM/2 プログラミングを楽しむには，上述の資料は必須です．

## Warp の オンラインドキュメント・セット

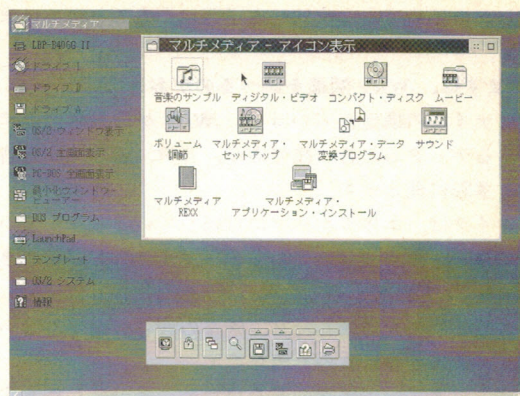
本稿では，Warp のオンラインドキュメント・セットを利用し，mm4emx10 の都合に合わせて，読み替えています．

このセットは，28 個の **.INF** ファイル(一つだけはテキストファイル)からなり，合計 16 M バイトの大きさです．五つが MMPM/2 関係のドキュメントです．**表6-1** にその内容を示します．

残念ながら，Warp は最新の OS ですから，筆者の手にしている mm4emx10 で，このすべてが利用できるわけではありません．

〈表6-1〉Warp のオンラインドキュメント・セットから

```
Multimedia Application Programming Guide
  Multimedia Application Programming Environment
  Media Control Interface (mci)
  Multimedia Logical Devices
  Amplifier-Mixer Device
  Waveform Audio Device
  Sequencer Device
  CD Audio Device
  CD-XA Device
  Videodisc Device
  Direct Interface Video Extensions (DIVE)
  Captioning
  OS/2 Multimedia Controls
  Multimedia I/O File Services
  Resource Interchange File Format (RIFF) Services
  Sample Application Programs
  Installing a Program Using MINSTALL
  Caption DLL
  High-Level Service API

Multimedia Programming Reference 1
  MCI Functions
    mciGetDeviceID()
    mciGetErrorString()
    mciQuerySysValue()
            :
            :
```

〈図6-2〉「マルチメディア」フォルダー

しかし，OS/2 のマルチメディアを楽しむには十分と思いますし，獲得した知識は，そのまま Warp でも通用します．

なお，Toolkit 2.1 の **MMPROG.INF** の項目を参考のため，**表6-2** に示します．

## どこから始めるか

筆者の場合，データベースを職業としているためか，マルチメディアには，どのような種類のファイルがあるのかが，真っ先に気になります．

おおまかには，**.WAV** が波形データ，**.MID** が MIDI データ(器楽)，**.AVI** が動画，**.FLI**/ **.FLC** がアニメーション，その他，といったところでしょうか．

もちろん，最も関心のある分野から調べればよいのです．MMPM/2 の場合，後述する **mciPlayFile** 関数や

〈表6-2〉Toolkit 2.1 のマルチメディアドキュメントの項目

```
Multimedia Applidation Programming Guide
  Multimedia Application Programming Environment
  Media Control Interface (mci)
  Multimedia Logical Devices
  Amplifier-Mixer Device
  Waveform Audio Device
  Sequencer Device
  CD Audio Device
  CD-XA Device
  Videodisc Device
  Direct Interface Video Extensions (DIVE)
  Captioning
  OS/2 Multimedia Controls
  Multimedia I/O File Services
  Resource Interchange File Format (RIFF) Services
  Sanple Application Programs
  Installing a Progran Using MINSTALL
  Caption DLL
  High-Level Service API

Multinedia Programming Reference 1
  MCI Functions
    mciGetDeviceID()
    mciGetErrorString()
    mciQuerySysValue()
```

ストリングコマンドは，各ファイルで共通ですから，どれから始めても，遠回りの学習になることはありません．

● まずは，.WAV 関連を調べるのがお薦め

ファイルの構造からいうと，**.WAV** が最も単純です．扱えるデータも，音一般であり，音声，音楽，効果音と，多彩に使えます．

マイクは，本体やボードに付属していることが多いので，録音/再生がユーティリティですぐにお試しできます．

自分で波形を合成したり，既存の音にとんでもないエフェクトをかけるのも簡単です．

ディジタルフィルタの本を買うと，当分楽しめます．係数を間違えると，カオスの音が聞けたりします(経験者は語る)．

もし迷っているのなら，波形データ，および **.WAV** ファイルを自由自在に操ることを目標とするのが，マルチメディア関係のシステムを理解する早道だと思います．

● MIDI データファイル .MID

音楽に関心がある方なら，**.MID** を調べると面白いでしょう．MIDI ファイルの形式については，本誌の 1995 年 4 月号まで連載があったので，参考にしていただければと思います．

非力な音源でも，上手なファイルでは，心地よい音楽が聴けてびっくりしたりします．互換性の障壁はあるにしても，MIDI 音源の高機能化は時間の問題でしょう．

音楽を情報伝達の手段として使うのは，筆者の記憶では「サイモン」というゲーム(音程の順にボタンを押すと，一面(?)クリア．知っている人は，間違いなくおじさん，おばさんか)が最後です．

しかし，音楽には多重の演奏が調和してしまうなど，興味深い性質があるので，研究に値するのではないでしょうか．

● アニメーションファイル .FLI/ .FLC

**.FLI**/ **.FLC** の形式の資料は，パソコン通信で手に入ります．基本的には，ランレングスの圧縮で，アニメ的な絵での圧縮率が良好です．

筆者は，この形式の(掟破りの)3D アニメを収録した CD-ROM(Walkthroughs and Flybys CD)を持っています．

**.FLI** は，320×200×256 色の VGA 画面専用の形式です．一瞬粗いように思えますが，100 インチ以上の大画面で，バックグラウンド・ミュージック付きで見ると，すさまじい迫力です．

**.FLC** は，640×480×256 色の画面用です．これも，ある程度以上の大きさの画面で見ると，その美しさに圧倒されます．

**〈リスト6-1〉 mciPlayFile( )のサンプルプログラム mmtest1.c**

```
/* mmtest1.c YOkada 950703 */

#define INCL_WIN
#define INCL_GPI
#include <os2.h>

#define INCL_OS2MM
#define INCL_MACHDR
#include <os2me.h>

MRESULT EXPENTRY
clientWndProc(HWND, ULONG, MPARAM, MPARAM);

int main() {
  HAB hab;
  HMQ hmq;
  HWND frame, client;
  QMSG qmsg;
  static ULONG fl = FCF_TITLEBAR | FCF_SYSMENU |
       FCF_SIZEBORDER | FCF_MINMAX | FCF_SHELLPOSITION
       | FCF_TASKLIST;
  static CHAR cclass[] = "mmtest1";
  ULONG rc;

  hab = WinInitialize(0);
  hmq = WinCreateMsgQueue(hab, 0);
  WinRegisterClass(hab, cclass, clientWndProc,
    CS_SIZEREDRAW | WS_ANIMATE, 0);
  frame = WinCreateStdWindow(HWND_DESKTOP,
          WS_VISIBLE, &fl, cclass, NULL, 0,
          NULLHANDLE, 0, &client);

  while (WinGetMsg(hab, &qmsg, 0, 0, 0)) {
    WinDispatchMsg(hab, &qmsg); }

  WinDestroyMsgQueue(hmq);
  WinTerminate(hab);
  return 0;
}

MRESULT EXPENTRY
clientWndProc(HWND hwnd, ULONG msg, MPARAM mp1,
                                    MPARAM mp2) {
  HPS hps;
  POINTL sp;

  switch (msg) {
  case WM_PAINT:
    hps = WinBeginPaint(hwnd, NULLHANDLE, NULL);
    sp.x = 20; sp.y = 20;
    GpiCharStringAt(hps, &sp, 15, "Click anywhere!");
    WinEndPaint(hps);
    return 0;
  case WM_BUTTON1DOWN:
    mciPlayFile(
      hwnd,
      "d:¥¥mmos2¥¥sounds¥¥boo.wav", // ドライブの指定
      MCI_ASYNC,
      NULL,
      NULLHANDLE);
    break;
  case WM_ERASEBACKGROUND:
    return (MRESULT)TRUE;
  }
  return WinDefWindowProc(hwnd, msg, mp1, mp2);
}
```

**〈リスト6-2〉 リスト6-1 の定義ファイル mmtest1.def**

```
NAME            MMTEST1 WINDOWAPI
DESCRIPTION     'MMTEST1'
STACKSIZE       32768
EXPORTS         clientWndProc
```

● **動画ファイル .AVI**

**.AVI** に関しては，現状のハードでの互換性や絵の品質もあり，関心はあるものの，積極的には調べていません．

しかし，DVDなどは大きな市場があると筆者も思いますし，そのうち，波形データ並みに，家庭でディジタル編集，などという日も来るのではないでしょうか．

● **その他**(リアルタイム再生)

もちろん，ゲームやバーチャルリアリティなど，リアルタイムでアニメしたり，音を発生させるプログラムも，マルチメディアと言えると思います．

ゲームマシンではもう常識でしょうが，パソコンでもマルチメディア用アクセラレータチップが普及してほしいと思います．

ウィンドウの次はバーチャルリアリティだ，などと言うと，いささか無責任でしょうか(ディズニーの映画，TRONの見過ぎか)．

## 高機能なマルチメディア関数

● **mciPlayFile**(リスト6-1，リスト6-2)

MMPM/2の中でも，最も簡単で強力な関数が，**mciPlayFile( )** です．

プログラムを動かし，クライアントウィンドウ内でクリックしてください．ブーイングが聞けるはずです(ドライブの指定は各自の環境に合わせて適当に変更する)．

▶ **mciPlayFile( )** の使い方

MMPM/2を使うためには，**os2me.h** をインクルードします．また，**INCL_OS2MM** を **#define** しておきます．

**INCL_MACHDR** は，ヘッダー内の **mciPlayFile( )** の宣言と，関連する **#define** を活性化します．

**mciPlayFile** の第1引数は，関連するメッセージを受け取るウィンドウのハンドルです．

第2引数は，演奏するファイル名です．

第3引数には，**mciPlayFile( )** の効果を指定します．演奏をストップしたり，いろいろあります．

**MCI_ASYNC** は，演奏を開始したらすぐに制御を戻す効果があります．**MCI_ASYNC** を指定しないと，演奏が終わるまで **mciPlayFile( )** が終わりません．

〈リスト6-3〉 mciPlayResource( )のサンプルプログラム mmtest2.c

```
/* mmtest2.c YOkada 950703 */

#define INCL_WIN
#define INCL_GPI
#include <os2.h>

#define INCL_OS2MM
#define INCL_MACHDR
#include <os2me.h>

MRESULT EXPENTRY
clientWndProc(HWND, ULONG, MPARAM, MPARAM);

int main() {
  HAB hab;
  HMQ hmq;
  HWND frame, client;
  QMSG qmsg;
  static ULONG fl = FCF_TITLEBAR | FCF_SYSMENU |
       FCF_SIZEBORDER | FCF_MINMAX | FCF_SHELLPOSITION
       | FCF_TASKLIST;
  static CHAR cclass[] = "mmtest2";
  ULONG rc;

  hab = WinInitialize(0);
  hmq = WinCreateMsgQueue(hab, 0);
  WinRegisterClass(hab, cclass, clientWndProc,
    CS_SIZEREDRAW | WS_ANIMATE, 0);
  frame = WinCreateStdWindow(HWND_DESKTOP, WS_VISIBLE,
       &fl, cclass, NULL, 0, NULLHANDLE, 0, &client);

  while (WinGetMsg(hab, &qmsg, 0, 0, 0)) {
    WinDispatchMsg(hab, &qmsg); }

  WinDestroyMsgQueue(hmq);
  WinTerminate(hab);
  return 0;
}

MRESULT EXPENTRY
clientWndProc(HWND hwnd, ULONG msg, MPARAM mp1,
                                    MPARAM mp2) {
  HPS hps;
  POINTL sp;

  switch (msg) {
  case WM_PAINT:
    hps = WinBeginPaint(hwnd, NULLHANDLE, NULL);
    sp.x = 20; sp.y = 20;
    GpiCharStringAt(hps, &sp, 15, "Click anywhere!");
    WinEndPaint(hps);
    return 0;
  case WM_BUTTON1DOWN:
    mciPlayResource(
      hwnd,
      NULLHANDLE, /* resource in .exe */
      RT_WAVE,
      10,    /* resource ID */
      MCI_ASYNC,
      NULL,
      NULLHANDLE);
    break;
  case WM_ERASEBACKGROUND:
    return (MRESULT)TRUE;
  }
  return WinDefWindowProc(hwnd, msg, mp1, mp2);
}
```

〈リスト6-4〉 リスト6-3の定義ファイル mmtest2.def

```
NAME          MMTEST2 WINDOWAPI
DESCRIPTION   'MMTEST2'
STACKSIZE     32768
EXPORTS       clientWndProc
```

〈リスト6-5〉 リスト6-3のリソースファイル mmtest2.rc

```
/* mmtest2.rc YOkada 950710 */

#define INCL_OS2MM
#include <os2me.h>

RESOURCE RT_WAVE   10    d:¥mmos2¥sounds¥boo.wav
```

第4，5引数は，.avi ファイルの再生時に使います．それぞれ，ウィンドウタイトル，表示するウィンドウのハンドルです．

つまり，**mciPlayFile( )** は，さまざまなマルチメディアファイルを再生できます．

● mciPlayResource（**リスト6-3，リスト6-4，リスト6-5**）

リソースにマルチメディアのデータを含ませることもできます．

前述の**リスト6-1**とほぼ同じで，操作法も変わりません．ただし，データはファイルでなく，リソースにあるのが違いです．

まず，**.rc** ファイルを作ります．**RT_WAVE** というキーワードを認識させるため，**os2me.h** をインクルードしないといけません．そのため，リソースコンパイラにインクルードファイルのある場所を教えます．

**rc -r -I d:¥emx¥include mmtest2.rc**

などとします．**mmtest2.res** ができあがるはずです．

Cのプログラムをコンパイルしたら，**.res** を **.exe** に含ませないといけません．例えば，コンパイル時に，

**gcc -o mmtest2.exe mmtest2.c mmtest2.def mmtest2.res -los2me -Zomf -Zmtd -s**

などと指定します．

**mciPlayResource** の第2引数は，リソースのある **.dll** ファイルを（間接的に）指定します．リソースが **.exe** にあるなら，**NULLHANDLE** です．

第3引数には，どのマルチメディアの種類かを指定します．

第4引数は，リソースの ID 番号です．

その他の引数は，**mciPlayFile( )** と同じです．

## mciSendString（リスト6-6）

**mciSendString( )** は，マルチメディア REXX でおなじみのストリングコマンドを実行する関数です．**mm4emx10** に付属する **mci** ディレクトリにサンプルがあります．

**リスト6-6** は，標準入力からストリングコマンドを読み込むので，実行には，

**mms < bach.mci**

などと，ストリングコマンドのあるファイルをリダイレクトします．**bach.mci** は，サンプルに入っているので，それをコピーするか，自分でエディタを使って作ります．

**mciSendString( )** の第1引数に，“**play ...**”といったストリングコマンドの文字列を指定します．

“**sysinfo all name 1**”などの情報を返すコマンドでは，バッファ（**CHAR** の配列）を用意し，そのアドレスを第2引数で渡します．バッファの大きさを第3引数で指定します．

“**play ... notify**”などと **notify** を指定すると，演奏が終わった時点で，第4引数に指定したウィンドウに，**MM_MCINOTIFY** メッセージが送られます．第5引数は，**MM_MCINOTIFY** メッセージのパラメータに送られる，**USHORT** 型のデータです．

## WAVE ファイルの構造

WAVE（**.wav**）ファイルは Windows 3.1 とまったく同じです．ファイルの構造を**図6-3**に示します．

**.wav** ファイルを作る簡単なプログラムを書いてみました（**リスト6-7**）．

**wav3**

と，プログラムを実行すると，**ttt.wav** が作られます．このファイルの内容を聴くには，OS/2 セッションで，

**play file=ttt.wav**

とします．

## プレイリスト（オルガンプログラムを作る）

メモリ上の波形データを演奏するには，プレイリストを用意し，**mciSendCommand( )** を使います．

**mciSendString( )** は，**mciSendCommand( )** に内部的に変換されているとのことですから，こちらが基本的な関数です．

プレイリストは，四つの **ULONG** からなる構造体の配列です．最初の **ULONG** が操作を示し，あとの三つの **ULONG** がパラメータです．

命令は九つあります．詳しくはドキュメントを見ていただくことになります．しかし，本稿で取り上げた，**DATA_OPERATION**（データの本体をポイントする），

〈リスト6-6〉 mciSendString( )のサンプルプログラム
**mms.c**

```
/* mms.c YOkada 950704 mciSendString */

#include <os2.h>

#define INCL_OS2MM
#include <os2me.h>

#include <stdio.h>

#define N 256

int main() {
  ULONG rc;
  unsigned char s[N];

  for (;;) {
    if (NULL == gets(s)) break;
    rc = mciSendString(
      s, NULL, 0, NULLHANDLE, 0);
  }
  return 0;
}
```

**BRANCH_OPERATION**（他のデータを演奏する），**EXIT_OPERATION**（プレイリストの最後）だけで，メモリ上のデータは扱えます．

**DATA_OPERATION** 命令の第1引数は，生のデータへのポインタです．**.wav** ファイルの正味のデータの形式で書いておきます．第2引数は，データの大きさです．

**BRANCH_OPERATION** の第2引数は，飛び先のプレイリストの添字です．1番目のプレイリストのデータは配列の先頭ですから，0となります．以下のプログラムでは，無音の無限ループを作るために用いています．

● **オルガンプログラム**

プレイリストを使って，簡単なオルガンのプログラムを作りました（**リスト6-8，リスト6-9，リスト6-10**）．

プログラムを動かすと，鍵盤と音色の操作卓のウィンドウが現れます（**図6-4**）．

鍵盤は，クロマチックハーモニカに合わせて，3オクターブとしました．鍵盤をクリックすると，音が出ます．音は，0.5秒間出ます．延ばすことはできません．音が出ている途中で，他の鍵盤を押すと，その音が0.5秒間出ます．

スピンボタンを操作すると，音色を変えることができます．設定を変更したら，Set! ボタンをクリックします．時計表示が消えたら，新しい音色が使えます．

「bias」とあるのは，音域の変更です．左端の「ド」の鍵盤の高さを決めます．MIDIを知っている方なら，ノートナンバーと言えばわかるでしょう．

〈図6-3〉WAVE（.wav）ファイルの構造

| バイト＼オフセット | 0 | 1 | 2 | 3 |
|---|---|---|---|---|
| 0 | 'R' | 'I' | 'F' | 'F' |
| 4 | 以下のファイルの大きさ（ファイルサイズ−8） | | | |
| 8 | 'W' | 'A' | 'V' | 'E' |
| 12 | 'f' | 'm' | 't' | 0x20 |
| 16 | 以下のヘッダーの大きさ（＝16） | | | |
| 20 | 01（形式） | | mono＝1/stereo＝2 | |
| 24 | 秒あたりのサンプル | | | |
| 28 | 秒あたりのバイト数 | | | |
| 32 | 1単位のバイト数 | | サンプルあたりのビット数 | |
| 36 | 'd' | 'a' | 't' | 'a' |
| 40 | 以下のデータの大きさ（ファイルサイズ−44） | | | |
| 44〜 | 以下はデータ．(b)〜(e)を参照 | | | |

(a) .wavファイルの先頭部

| | |
|---|---|
| 0 | データ0 |
| 1 | データ1 |
| ⋮ | ⋮ |

(b) モノラルで8ビット（**0x80** が無信号）

| | |
|---|---|
| 0 | データ0（左） |
| 1 | データ0（右） |
| 2 | データ1（左） |
| ⋮ | ⋮ |

(c) ステレオで8ビット

| | |
|---|---|
| 0 | データ0（下位） |
| 1 | データ0（上位） |
| 2 | データ1（下位） |
| ⋮ | ⋮ |

(d) モノラルで16ビット（**0x0000** が無信号）

| | |
|---|---|
| 0 | データ0（下位）（左） |
| 1 | データ0（上位）（左） |
| 2 | データ0（下位）（右） |
| 3 | データ0（上位）（右） |
| 4 | データ1（下位）（左） |
| ⋮ | ⋮ |

(e) ステレオで16ビット

〈リスト6-7〉.wavファイルを作るサンプルプログラム
**wav3.c**

```
/* wav3.c YOkada 950703 .wav create */

#define INCL_DOS

#include <os2.h>
#include <math.h>

#define BL (44100 * 18)

static unsigned short b[BL + 22];

int main() {
  unsigned short *bx;
  double pi2, log2, fq, x, y;
  static PSZ fname = "ttt.wav";
  HFILE hfile;
  ULONG ac;
  int i;

  pi2 = 8.0 * atan(1.0);
  log2 = log(2.0);
  b[0] = 'R' + 'I' * 256; b[1] = 'F' + 'F' * 256;
  b[2] = (BL * 2 + 9 * 4) & 0xFFFF ;
  b[3] = (BL * 2 + 9 * 4) >> 16;
  b[4] = 'W' + 'A' * 256; b[5] = 'V' + 'E' * 256;
  b[6] = 'f' + 'm' * 256; b[7] = 't' + ' ' * 256;
  b[8] = 16; b[9] = 0;
  b[10] = 1; b[11] = 1; /* mono */
  b[12] = 44100; b[13] = 0;
  b[14] = (44100 * 2) & 0xFFFF;
  b[15] = (44100 * 2) >> 16;
  b[16] = 2; b[17] = 16;
  b[18] = 'd' + 'a' * 256; b[19] = 't' + 'a' * 256;
  b[20] = (BL * 2) & 0xFFFF; b[21] = (BL * 2) >> 16;

  bx = b + 22; x = 0.0;
  for (i = 0; i < BL; i++) {
    fq = exp((double)i / (44100.0*3.0) * log2) * 64.0;
    x += pi2 * fq / 44100.0;
    y = sin(x);
    *bx = (int)(16382.0 * y);
    bx++;
  }
  DosBeep(1000, 100);
  DosOpen(fname, &hfile, &ac, 0, FILE_NORMAL,
          OPEN_ACTION_REPLACE_IF_EXISTS |
          OPEN_ACTION_CREATE_IF_NEW,
          OPEN_ACCESS_WRITEONLY |
          OPEN_SHARE_DENYNONE,
          NULL);
  DosWrite(hfile, b, BL * 2 + 11 * 4, &ac);
  DosClose(hfile);
  return 0;
}
```

中央の「ド」を60とし，半音高いと1つ数字が増えます．設定できるのは，24から72です．

「cut(x10)」とあるのは，ローパスフィルタのカットオフ周波数の設定です．10のとき，鍵盤の音と同じ周波数でカットオフします．20なら，2倍の周波数です．2次のアクティブフィルタをシミュレートしています．音源は正弦波の合成で，のこぎり波をシミュレートしています．

「Q(x10)」とあるのはいわゆる$Q$で，カットオフ周波数でのピークの高さを指定します．10なら1倍，ということです．

cutもQも，数字が増えると倍音が増えるので，音の輝きが増します．あえて表現すると，cutは複雑さが増し，Qは癖のある音になります．

### ● プログラムの解説

#### ▶ make_wave( ) ～波形データを作る関数～

**make_wave( )** は，各鍵盤に対応する，37の波形データを作る関数です．**aa[ ]** は，32倍音までの各倍音成分の大きさの配列です．波形の一周期が配列 **vv[ ]** にセットされます．最終的には，**wav[ ][ ]** の中身が作られます．

音は，急に立ち上がり，0.25秒間持続し，0.25秒かけて減衰します．この音の大きさの推移，つまりエンベロープは聴感上の音色に大きな影響を与えるので，興味のある方は，いろいろお試しください．

この延べ18.5秒のデータを作るのに，486DX4-100 MHzで3秒ほどかかります．

#### ▶ key_no( ) ～鍵盤位置の座標を計算する関数～

ウィンドウ上の鍵盤はボタンでも何でもなく，単に線が書いてあるだけです．マウスのボタンがクリックされたら，座標からどの鍵盤が押されたかを，いちいち計算しています．その計算部分が，**key_no( )** です．

#### ▶ main( )

(1) プレイリストの作成とオープン

メインウィンドウを作成したら，メッセージループに入る前に，プレイリストを作成し，オープンしています．

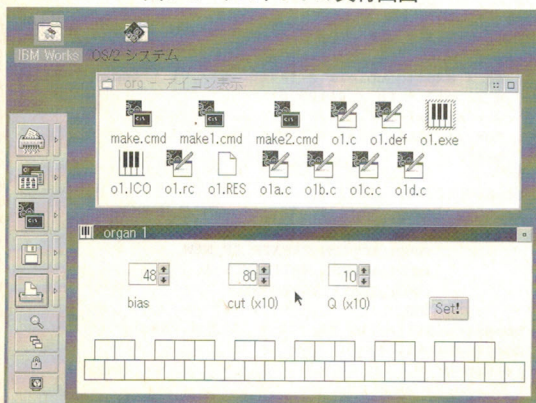
(2) **mciSendCommand( )**

**mciSendCommand( )** の第1引数は，デバイスIDです．このIDは，オープンのときに得られます．第2引数は，コマンドメッセージです．最初は，**MCI_OPEN** を指定します．第3引数はフラグです．プレイリストを使うには，**MCI_OPEN_PLAYLIST** を指定します．第4引数は，メッセージのパラメータです．コマンドメッセージによって型が決められています．

例えば，**MCI_OPEN** コマンドでは，**MCI_OPEN_PARAMS** 型となります．この型の **pszElementName**

〈図6-4〉リスト6-8の実行画面

〈リスト6-8〉プレイリストを使ったオルガンプログラム o1.c

```
/* o1.c YOkada 950709 electric organ 1 */

#define INCL_WIN
#define INCL_DOS
#define INCL_GPI
#include <os2.h>

#define INCL_OS2MM
#include <os2me.h>
#include <math.h>

#define BL (44100 / 2)

HWND hwnds[7];

static int nn = 48;
static double cut = 4.0;
static double q = 1.0;
static unsigned short wav[37][BL];
static unsigned short wav0[4] = {0,0,0,0};

static MCI_OPEN_PARMS mop;

typedef struct {
  ULONG ope;  ULONG op1;
  ULONG op2;  ULONG op3;
} PL1;

static PL1 pl[4];

MRESULT EXPENTRY
clientWndProc(HWND, ULONG, MPARAM, MPARAM);

void make_wave() {
  int i, j;
  unsigned short *bi;
  double pi2, log2;
  double x, dx, v, dv, fq;
  double aa[32], vv[1024];

  pi2 = 8.0 * atan(1.0);
  log2 = log(2.0);
  for (i = 1; i <= 32; i++) {
    aa[i - 1] = 1 / (double)i; /* saw wave */
    x = (double)i * (double)i / (cut * cut);
    aa[i - 1] *= 1 / sqrt(1.0 + (((1.0 / (q * q) -
                                2.0) + x) * x));
  }
  for (i = 0; i < 1024; i++){
    vv[i] = 0.0;
    for (j = 0; j < 32; j++) {
      vv[i] += aa[j] * sin((double)(j + 1) * pi2 *
                            (double)i / 1024.0);
    }
  }
  for (i = 0; i < 37; i++) {
    fq = 440.0 * exp((log2 / 12.0) *
                      ((double)(nn + i) - 69.0));
```

メンバは，型は**PSZ**(つまり文字列へのポインタ)ですが，セットするのはプレイリストのポインタです．

データの形式(ステレオかモノラルかなど)は，**MCI_SET** コマンドメッセージでセットします．

(3)スピンボタン

OS/2では，スピンボタンが標準で使えます．スピンボタンは，上下矢印ボタンの付いた，エディットボックスです．値を選ぶのに使います．

取り得る値は，上下限のある整数で指定するか，文字列の配列で指定します．ここでは簡単のため，整数を指定しています．

スピンボタンを生成するには，第2引数に **WC_SPINBUTTON** を指定した，**WinCreateWindow( )** を使います．上下限などは，**WinSendMsg( )** で，スピンボタンにメッセージを送ることによって設定します．ここでは，上下限と初期値を設定しています．

▶ **clientWndProc( )**

ウィンドウプロシージャでは，**WM_PAINT** とマウスのクリック，ボタンの選択に応答しています．

(1) **WM_PAINT**

**WM_PAINT** では鍵盤を描いています．要するに，ただの絵です．

(2) **WM_BUTTON1DOWN**

マウスの左ボタンが押されたらここに来ます．鍵盤

〈リスト6-8〉プレイリストを使ったオルガンプログラム **o1.c**(つづき)

```
      v = 2048.0; dv = v / (double)(BL / 2);
      x = 0.0; dx = fq / 44100.0 * 1024.0;
      bi = wav[i];
      for (j = 0; j < BL; j++) {
        if (BL - j < BL / 2) v -= dv;
        *bi++ = (int)(v * vv[(int)x & 0x3ff]);
        x += dx;
      }
    }
  }

  int key_no(short x, short y) {
    static int up[23] =
      {1, 3, -1, 6, 8, 10, -1, 13, 15, -1, 18, 20, 22,
       -1, 25, 27, -1, 30, 32, 34, -1, -1, -1};
    static int dn[23] =
      {0, 2, 4, 5, 7, 9, 11, 12, 14, 16, 17, 19, 21, 23,
       24, 26, 28, 29, 31, 33, 35, 36, -1};
    if (y < 16) return -1;
    if (y < 16 + 24) {
      if (x < 8) return -1;
      else return dn[(x - 8) / 24]; }
    if (y < 16 + 48) {
      if (x < 20) return -1;
      else return up[(x - 20) / 24]; }
    return -1;
  }

  int main() {
    HAB hab;  HMQ hmq;
    static CHAR cclass[] = "o1";
    HWND frame, client;
    RECTL rcl;  QMSG qmsg;
    static ULONG fl = FCF_TITLEBAR | FCF_SYSMENU |
                      FCF_MINBUTTON | FCF_BORDER |
                      FCF_SHELLPOSITION | FCF_TASKLIST |
                      FCF_ICON;
    MCI_WAVE_SET_PARMS wsp;
    MCI_GENERIC_PARMS mgp;
    hab = WinInitialize(0);
    hmq = WinCreateMsgQueue(hab, 0);
    make_wave();
    WinRegisterClass(hab, cclass, clientWndProc,
      CS_SIZEREDRAW, 0);
    frame =
       WinCreateStdWindow(HWND_DESKTOP, 0, &fl, cclass,
      "organ 1", 0, NULLHANDLE, 1, &client);

    rcl.xLeft = 0; rcl.xRight = 22 * 24 + 16;
    rcl.yBottom = 0; rcl.yTop = 2 * 24 + 16 + 120;
    WinCalcFrameRect(frame, &rcl, FALSE);
    WinSetWindowPos(frame, HWND_TOP, 0, 0,
                    rcl.xRight - rcl.xLeft,
                    rcl.yTop - rcl.yBottom,
                    SWP_SIZE | SWP_MOVE | SWP_SHOW);
    pl[0].ope = DATA_OPERATION;
    pl[0].op1 = (ULONG)wav[0];
    pl[0].op2 = BL * 2; pl[0].op3 = 0;
    pl[1].ope = DATA_OPERATION; pl[1].op1 = (ULONG)wav0;
    pl[1].op2 = 4 * 2; pl[1].op3 = 0;
    pl[2].ope = BRANCH_OPERATION; pl[2].op2 = 1;
    pl[3].ope = EXIT_OPERATION;
    mop.hwndCallback = NULLHANDLE;
    mop.usDeviceID = 0;
    mop.pszDeviceType = MCI_DEVTYPE_WAVEFORM_AUDIO_NAME;
    mop.pszElementName = (PSZ)pl;
    mciSendCommand(0, MCI_OPEN,
      MCI_WAIT | MCI_OPEN_PLAYLIST, &mop, 0);
    wsp.hwndCallback =NULLHANDLE;
    wsp.usChannels = 1;
    wsp.ulSamplesPerSec = 44100;
    wsp.usBitsPerSample = 16;
    mciSendCommand(mop.usDeviceID, MCI_SET,
                   MCI_WAIT | MCI_WAVE_SET_CHANNELS |
                   MCI_WAVE_SET_SAMPLESPERSEC |
                   MCI_WAVE_SET_BITSPERSAMPLE,
                   &wsp, 0);
    hwnds[0] = WinCreateWindow(frame, WC_STATIC, "bias",
      WS_VISIBLE | SS_TEXT, 60, 90, 100, 30,
      frame, HWND_TOP, 1000, NULL, NULL);
    hwnds[1] = WinCreateWindow(frame, WC_STATIC,
                                            "cut (x10)",
      WS_VISIBLE | SS_TEXT, 180, 90, 100, 30,
      frame, HWND_TOP, 1001, NULL, NULL);
    hwnds[2] = WinCreateWindow(frame, WC_STATIC,
                                            "Q (x10)",
      WS_VISIBLE | SS_TEXT, 300, 90, 100, 30,
      frame, HWND_TOP, 1002, NULL, NULL);
    hwnds[3] = WinCreateWindow(client, WC_SPINBUTTON,
                NULL, WS_VISIBLE | SPBS_MASTER |
                SPBS_NUMERICONLY | SPBS_JUSTRIGHT,
                60, 130, 50, 40,
                client, HWND_TOP, 1003, NULL, NULL);
    WinSendMsg(hwnds[3], SPBM_SETLIMITS, MPFROMLONG(72),
                MPFROMLONG(24));
    WinSendMsg(hwnds[3], SPBM_SETCURRENTVALUE,
                MPFROMLONG(48), MPFROMLONG(0));
    hwnds[4] = WinCreateWindow(client, WC_SPINBUTTON,
                NULL, WS_VISIBLE | SPBS_MASTER |
                SPBS_NUMERICONLY | SPBS_JUSTRIGHT,
                180, 130, 50, 40, client, HWND_TOP,
                1004, NULL, NULL);
    WinSendMsg(hwnds[4], SPBM_SETLIMITS, MPFROMLONG(320),
                                         MPFROMLONG(1));
    WinSendMsg(hwnds[4], SPBM_SETCURRENTVALUE,
                    MPFROMLONG(80), MPFROMLONG(0));
    hwnds[5] = WinCreateWindow(client, WC_SPINBUTTON,
                NULL, WS_VISIBLE | SPBS_MASTER |
                SPBS_NUMERICONLY | SPBS_JUSTRIGHT,
                300, 130, 50, 40,
                client, HWND_TOP, 1005, NULL, NULL);
    WinSendMsg(hwnds[5], SPBM_SETLIMITS, MPFROMLONG(100),
                MPFROMLONG(5));
    WinSendMsg(hwnds[5], SPBM_SETCURRENTVALUE,
                MPFROMLONG(10), MPFROMLONG(0));
    hwnds[6] = WinCreateWindow(client, WC_BUTTON, "Set!",
```

が押されたかどうかを判定し，押されていたら，波形データの配列をプレイリストに設定し，**MCI_PLAY** コマンドメッセージを **mciSendCommand( )** しています．

ここではあまり意味がありませんが，メッセージがあればこのウィンドウプロシジャのウィンドウに通知させるように **hwndCallback** に **hwnd** を指定しています．マウスのクリックは，**WinDefWindowProc( )** に知らせるべきなので，**break ;** を使っています．

(3) **WM_COMMAND**

Set! ボタンの処理です．スピンボタンの数値を得るため，**WinSendDlgItemMsg( )** を使っています．**WinSendMsg( )** でもかまわないのですが，引数が使いやすいので使ってみました．

第2引数は，**WinCreateWindow( )** で指定した，スピンボタンのIDです．第3引数はメッセージ，第4引数，第5引数はパラメータです．マウスポインタを時計にし，**make_wave( )** を起動しています．

● まとめ

以上，本稿で取り上げた関数で，一通りのオーディオのプログラミングは楽しめるのではないかと思います．

ノートブックではオーディオはまだまだでしょう．しかし，デスクトップでは音源ボードがあたりまえになってきていますし，なくても少し余裕があれば，購入可能です．

特に，**.wav** ファイルは，アミューズメントだけでなく，音ならではの情報伝達に役立ちます．

ハードディスク・ドライブの容量も，CD-ROMなみになってきました．今こそ，音源を生かした応用プログラムが模索される時代だと思います．

〈リスト6-8〉プレイリストを使ったオルガンプログラム o1.c (つづき)

```
              IBLE | BS_PUSHBUTTON, 90, 50, 30,
              t, HWND_TOP, 1006, NULL, NULL);
  while (WinGetMsg(hab, &qmsg, NULLHANDLE, 0, 0)) {
    WinDispatchMsg(hab, &qmsg); }
  mgp.hwndCallback = NULLHANDLE;
  mciSendCommand(mop.usDeviceID, MCI_CLOSE,  MCI_WAIT,
                                               &mgp, 0);
  WinDestroyMsgQueue(hmq);
  WinTerminate(hab);
  return 0;
}

MRESULT EXPENTRY
clientWndProc(HWND hwnd, ULONG msg, MPARAM mp1,
                                     MPARAM mp2) {
  HPS hps;  POINTL ptl;  int i, nn1;
  MCI_PLAY_PARMS mpp;
  MCI_GENERIC_PARMS mgp;

  switch (msg) {
  case WM_PAINT:
    hps = WinBeginPaint(hwnd, NULLHANDLE, NULL);
    ptl.x = 8; ptl.y = 16; GpiMove(hps, &ptl);
    ptl.x = 8 + 22 * 24; GpiLine(hps, &ptl);
    ptl.x = 8; ptl.y = 16 + 24; GpiMove(hps, &ptl);
    ptl.x = 8 + 22 * 24; GpiLine(hps, &ptl);
    for (i = 0; i <= 22; i++) {
      ptl.x = 8 + 24 * i; ptl.y = 16;
      GpiMove(hps, &ptl); ptl.y = 16 + 24;
      GpiLine(hps, &ptl);
    }
    for (i = 0; i <= 2; i++) {
      ptl.x = 20 + 7 * 24 * i; ptl.y = 16 + 48;
      GpiMove(hps, &ptl);
      ptl.x = 20 + 2 * 24 + 7 * 24 * i;
      GpiLine(hps, &ptl);
      ptl.x = 20 + 3 * 24 + 7 * 24 * i;
      ptl.y = 16 + 48;
      GpiMove(hps, &ptl);
      ptl.x = 20 + 6 * 24 + 7 * 24 * i;
      GpiLine(hps, &ptl);
    }
    for (i = 0; i <= 20; i++) {
      ptl.x = 20 + 24 * i; ptl.y = 16 + 24;
      GpiMove(hps, &ptl);  ptl.y = 16 + 48;
      GpiLine(hps, &ptl);
    }
    WinEndPaint(hps);
    return 0;
  case WM_BUTTON1DOWN:
    nn1 = key_no(SHORT1FROMMP(mp1),
                         SHORT2FROMMP(mp1));
    if (nn1 < 0) break;
    pl[0].op1 = (ULONG)wav[nn1];
    mpp.hwndCallback = hwnd;
    mciSendCommand(mop.usDeviceID,
      MCI_PLAY, 0, &mpp, 0);
    break;
  case WM_COMMAND:
    i = SHORT1FROMMP(mp1);
    if (i != 1006) return 0;
    WinSendDlgItemMsg(hwnd, 1003, SPBM_QUERYVALUE,
     MPFROMP(&nn), MPFROM2SHORT(0,SPBQ_ALWAYSUPDATE));
    WinSendDlgItemMsg(hwnd, 1004, SPBM_QUERYVALUE,
     MPFROMP(&i), MPFROM2SHORT(0, SPBQ_ALWAYSUPDATE));
    cut = (double)i / 10.0;
    WinSendDlgItemMsg(hwnd, 1005, SPBM_QUERYVALUE,
      MPFROMP(&i),
       MPFROM2SHORT(0, SPBQ_ALWAYSUPDATE));
    q = (double)i / 10.0;
    WinSetPointer(HWND_DESKTOP,
      WinQuerySysPointer(HWND_DESKTOP,
                           SPTR_WAIT, FALSE));
    make_wave();
    WinSetPointer(HWND_DESKTOP,
               WinQuerySysPointer(HWND_DESKTOP,
                           SPTR_ARROW, FALSE));
    return 0;
  case WM_ERASEBACKGROUND:
    return (MRESULT)TRUE;
  }
  return WinDefWindowProc(hwnd, msg, mp1, mp2);
}
```

〈リスト6-9〉リスト6-8の定義ファイル o1.def

```
NAME           O1 WINDOWAPI
DESCRIPTION    'o1.def YOkada 950709'
HEAPSIZE       65536
STACKSIZE      65536
EXPORTS        clientWndProc
```

〈リスト6-10〉リスト6-8のリソースファイル o1.rc (o1.ico は各自で作る)

```
/* o1.rc YOkada 950709 */

ICON   1 o1.ico
```

# Excel 5.0のVBA活用マニュアル

## 第9回　文字列やフォントの処理

西田雅昭

セル内の文字を操作する際には，二つのオブジェクトを操作します．一つは**Characters**オブジェクトです．これは，文字どおり文字列です．一般の文字列操作を行う場合には，この**Characters**オブジェクトを処理します．**Characters**オブジェクトは，セル内の文字列だけでなく，

**AxisTitle，Button，Buttons，ChartTitle，CheckBox, CheckBoxes, DataLabel, DialogFrame, DrawingObjects，DropDown，DropDowns，EditBox，EditBoxes, GroupBox, GroupBoxes, Label, Labels, OptionButton, OptionButtons, Range, TextBox, TextBoxes**

などのオブジェクトから呼び出すことができます．

二つ目は，**Font**オブジェクトです．こちらは文字の属性を操作する際に使います．**Font**オブジェクトも，セルだけでなく，

**AxisTitle，Button，Buttons，Characters，ChartArea，ChartTitle，DataLabel，DataLabels，DrawingObjects, GroupObject, GroupObjects, Legend, LegendEntry，PlotArea，Range，Style，TextBox，TextBoxes，TickLabels**

などで呼び出すことができます．

### 文字列の取得と訂正

セル内の文字列を取得する際には，**Characters**メソッドを使います．

```
Range("A1").Characters
```

実際のようすを画面で示したいのですが，**Characters**オブジェクトは，それだけでは**Print**コマンドやイミディエイトペインで見ることができません．そこで，**Caption**プロパティを使います(図9-1)．

```
? Range("A1").Characters.Caption
abcdefg
```

**【参考】**

**Caption**プロパティは，いろいろなオブジェクトのタイトルやキャプションの操作に使えます．

おもしろい例を示しましょう．カスタムなシステムを作る際，Excelのメインウィンドウのタイトルを書き直したいときがあります．このような場合には，

```
Application.Caption = "トラコン用"
```

とすると，タイトルを書き換えることができます(図9-2)．本来のタイトルに戻す場合には，

```
Application.Caption = Empty
```

としてください．ただし，Macintoshでは内容の取得はできますが，内容の訂正はできません．

もちろん，他のウィンドウでも同じことができます．

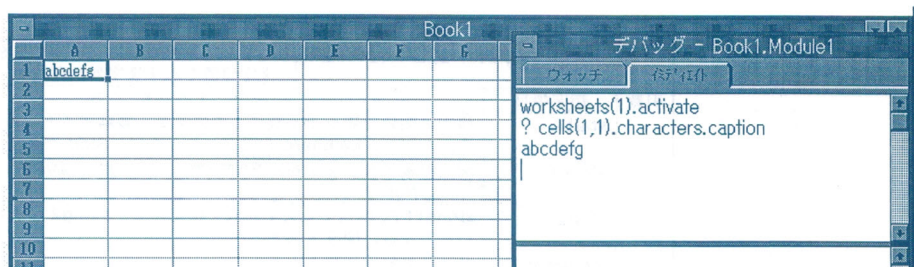
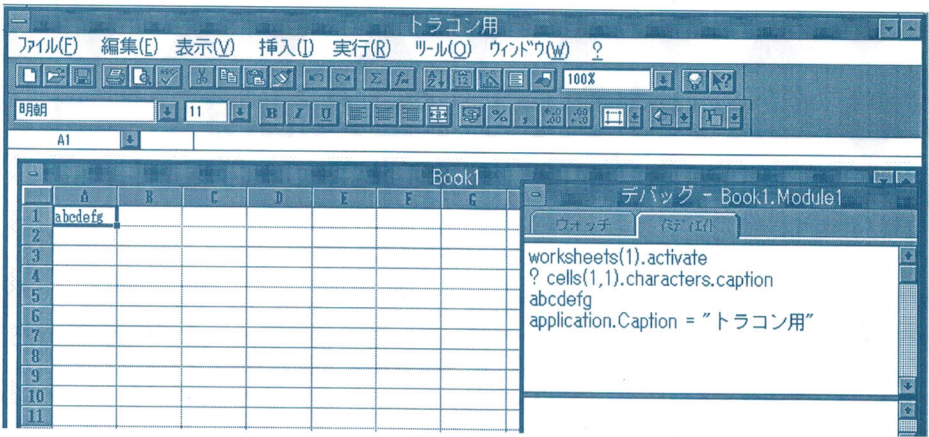
実は単純に，

```
? Range("A1")
abcdef
```

としても，セルの内容を見ることができます．これは，自動的に**Value**プロパティを呼び出すからのようです．

**Characters**メソッドで，**Characters**オブジェクト

〈図9-1〉
**Charactersオブジェクトでセル内の文字列を取得する例**

```
worksheets(1).activate
? cells(1,1).characters.caption
abcdefg
```

〈図9-2〉
**Caption プロパティでタイトルを書き換える例**

を取得する際には，取得する文字列の位置を指定することができます．

**? Range("A1").Characters(2, 3).Caption**

**bcd**

**Caption** プロパティを使うと，セル内の文字列を書き直すこともできます．

**? Range("A1").Characters.Caption = "CQ 出版"**

この2バイト文字の場合は，さきほどの部分文字列の取得はどうなるのでしょう．

**? Range("A1").Characters(2, 3).Caption**

**Q 出版**

Excel は，2バイト文字でも1バイト文字でも，文字数で処理をしてくれるのです．

**Characters** オブジェクトで，セル内のすべての文字列を書き直すのでは，**formula** プロパティと同じですが，部分文字列を書き直す場合に威力を発揮します．

**Range("A1").Characters(2, 1).Caption = "A"**

**? Range("A1").Characters.Caption**

**CA 出版**

**Range("A1").Characters(4, 1).Caption = "版社"**

**? Range("A1").Characters.Caption**

**CA 出版社**

## 文字列の処理

**Characters** オブジェクトで文字列の処理を行う場合，二つのメソッドが使用可能です．文字列を削除する **Delete** メソッドと，文字の前に文字列を挿入する **Insert** メソッドです．

実験の前に，

**Range("A1").Characters.Caption = "abcdef"**

とやっておきましょう．

**Delete** メソッドは，指定した文字列を消去します．

**Range("A1").Characters(2, 3).Delete**

**? Range("A1").Characters.Caption**

**aef**

反対に，**Insert** メソッドは(Help では)「指定した文字の前に文字列を挿入します」ということになっているのですが，実際は指定した文字列を置き換えるようです．

**Range("A1").Characters.Caption = "aef"**

**Range("A1").Characters(2, 1).Insert("bcd")**

**? Range("A1").Characters.Caption**

**abcdf**

といって，文字列の長さを指定しないで，

**Range("A1").Characters.Caption = "aef"**

**Range("A1").Characters(2).Insert("bcd")**

とやると，

**? Range("A1").Characters.Caption**

**abcd**

となってしまいます．これは，**Characters** メソッドの2番目の引数を省略した場合に，指定した文字の後ろ

〈表9-1〉
**InStr 関数の書式**

| InStr([start,]string1, string2[,compare]) | | ある文字列の中から別の文字列を検索し，最初に見つかった文字位置を返す文字列処理関数 |
|---|---|---|
| 引数 | 指定項目 | 内　容 |
| | **start** | 検索の開始位置を示す数式を指定する．省略すると，検索は最初の文字から開始する．ただし，**compare** を指定したときは，**Start** が必要． |
| | **string1** | 検索対象となる文字列式を指定する．**string1** に適切でないデータが含まれていたときは，**Null** を返す． |
| | **string2** | **string1** 内で検索する文字列式を指定する．**string2** に適切でないデータが含まれていたときは，**Null** を返す． |
| | **compare** | 文字列比較の比較モード<br>1 ………………テキストモード(大文字と小文字を区別しない)<br>0(既定値) ……バイナリモードの比較 |

〈図9-3〉 ボールドイタリックに指定

〈図9-4〉 下線の違い

〈図9-5〉 最初の文字だけ太くする

のすべての文字になってしまうわけです．

そこで，

```
Range("A1").Characters.Caption = "aef"
Range("A1").Characters(2, 0).Insert("bcd")
? Range("A1").Characters.Caption
abcdef
```

のように，指定する文字列の長さを0にすると，通常の言語の**Insert**コマンドと同じような動作になります．ちょっと面倒ですが，一般にいう**Replace**コマンドだと考えれば，かえって便利なメソッドかもしれません．

部分文字列の処理をするには，いくつかのプロパティを知っていると，細かい処理が可能になります．

まず，**Characters**オブジェクトの文字数を知る**count**プロパティです．

```
Range("A1").Characters.Caption = "abcde"
? Range("A1").Characters.count
5
Range("A2").Characters.Caption = "CQ出版"
? Range("A2").Characters.count
4
```

先ほどと同じように，1バイト，2バイトの区別なく，文字数を正確に知ることができます．これを利用すると，内容のわからない文字列を処理することも可能になります．

```
Range("A2").Characters(Characters.Count+1).
Insert(" 御中")
? Range("A2").Characters.Caption
CQ出版 御中
```

ここまで来ると，文字列の内部の検索もしたくなりますが，そのためのメソッドはありません．しかし，Basicですから Basicの関数を使うことができます．文字列検索用の関数**InStr( )**を使ってみましょう(**表9-1**)．

文字列処理関数を使うときには，さきほどの表示と同じように，**Caption**プロパティを利用するか，暗黙の**Value**プロパティを使ってください．

```
Range("A1").Characters.Caption = "abcde"
? InStr(Range("A1").Characters.Caprion, "d")
4
```

または，

```
Range("A1").Characters.Caption = "abcde"
? InStr(Range("A1"), "d")
4
```

VBAには，この他にたくさんの文字列処理関数があります．**表9-5**の「Basicの文字列処理関数」を参考にしてください．これらの関数の使い方は，Helpを参照してください．

## ワークシート関数について

Excelはまことに多機能で，ワークシート上で使える多くの関数を持っています．これをワークシート関数と呼びます．**表9-4**の「文字列操作関数(ワークシート)一覧」に，文字列操作に関するものを列挙しておきました．このワークシート関数もコードの上で使うことができます．ただし，このワークシート関数を使う際には，

```
? Application.Log(10)
1
```

のように，関数名の前に「**Application.**」をつけます．もし，これをつけないと

```
? log(10)
2.30258509299405
```

のように，Basicの関数になってしまいます(前者は10を底とする対数，後者はeを底とする自然対数)．

不思議なことに，筆者のExcelでは文字列処理のワークシート関数は，VBAの上ではエラーになってしまいます．ワークシート関数で，使えるものと使えないものがあるようです．現在原因を調べています．

## フォントの処理

フォントオブジェクトを取得するには，**Font**プロパティを使います．フォントオブジェクトには多くの属性がありますから，自由に設定することができます(**図9-3**)．

```
With Range("A1").Font
    .Bold = True
    .Italic = True
```

```
    .Size = 16
End with
```

また，**Font.Strikethrough**，**Font.Underline**，**Font.Subscript**，**Font.Superscript** などの飾りも可能です．下線が二通りあるので，この違いを次に示します．

```
Range("A1").Font.Underline = xlSingle
Range("B1").Font.Underline = xlSingleAccounting
Range("A3").Font.Underline = xlDouble
Range("B1").Font.Underline = xlDoubleAccounting
```

よく見ると，下線の位置が少し違います(図9-4)．

フォント名を指定する場合には，**FontStyle** プロパティを使いますが，この場合はフォント名を文字列として，正確に記述しなければなりません．

```
Range("A1").Font.FontStyle = "MS 明朝"
```

**Font** オブジェクトのプロパティとしては，この他に **Color** プロパティと **ColorIndex** がありますが，色に関しては，前号で詳しく説明しましたので割愛します．

## 文字ごとのフォントの指定

フォントの指定の際に，**Characters** オブジェクトを利用すると，文字列の中で特定の文字だけのフォント

〈表9-2〉Characters オブジェクトと Font オブジェクトとプロパティとメソッド

| 区分 | 種別 | 名前 | 取得・設定 / 引数 | | 説明 |
|---|---|---|---|---|---|
| | | **Characters** オブジェクト | | | セル，テキストボックス，ユーザー定義ボタンのいずれかに含まれる文字のコレクション(1文字以上の文字列)． |
| オブジェクトの取得 | m | **object.Characters(start, length)** | | | オブジェクトのテキストから，ある範囲の文字列(**Characters**)を返す |
| | | 引数 | **start** | | 返す最初の文字が何番目かを指定<br>省略すると，文字列の最初の文字から始まる範囲の文字 |
| | | | **length** | | 返す文字数を指定<br>省略すると，**start** に指定した文字から後のすべての文字 |
| | p | **object.Caption** | gs | | オブジェクトの持つ文字列 |
| | p | **Characters.Count** | g | | **Characters** に含まれる要素の数 |
| | m | **Characters.Delete** | | | 文字列を削除 |
| | m | **Characters.Insert(string)** | | | 選択した文字の前に文字列を挿入 |
| | | 引数 | **string** | | 挿入する文字列 |
| | p | **Characters.Font** | gs | | 文字列の中のフォント |
| | p | **Characters.Text** | gs | | 指定範囲の文字列 |
| | p | **Characters.Parent** | g | | オブジェクト |
| | | **Font** オブジェクト | | | フォント．フォントの持つすべての属性(フォント名，フォントサイズ，色など)をプロパティとして処理できる． |
| オブジェクトの取得 | p | **object.Font** | gs | | **Font** オブジェクトを返す |
| | | (**AxisTitle**，**Button**，**Buttons**，**Characters**，**ChartArea**，**ChartTitle**，**DataLabel**，**DataLabels**，**DrawingObjects**，**GroupObject**，**GroupObjects**，**Legend**，**LegendEntry**，**PlotArea**，**Range**，**Style**，**TextBox**，**TextBoxes**，**TickLabels**) | | | |
| 文字の種類 | p | **Font.Bold** | gs | | **True** ＝ 太字 |
| | p | **Font.Italic** | gs | | **True** ＝ 斜体 |
| | p | **Font.OutlineFont** | gs | | **True** ＝ アウトラインフォント(Windows では無効) |
| 色と形 | p | **Font.FontStyle** | gs | | フォントスタイルの値(文字列) |
| | p | **Font.Color** | gs | | フォントの色(RGB) |
| | p | **Font.ColorIndex** | gs | | フォントの色(カラーパレットのインデックス)<br>**xlAutomatic** 自動設定 |
| | p | **Font.Background** | gs | | テキストの背景の種類(グラフ上のテキストのみ)<br>**xlAutomatic**，**xlOpaque**，**xlTransparent** |
| | p | **Font.Shadow** | gs | | **True** ＝影付きフォント(Windows では無効) |
| | p | **Font.Size** | gs | | フォントのサイズ |
| 飾り | p | **Font.Strikethrough** | gs | | **True** ＝ 取り消し線を付ける |
| | p | **Font.Underline** | gs | | 下線の種類<br>**xlNone** ………下線なし<br>**xlSingle** ………下線<br>**xlDouble** ……二重下線<br>**xlSingleAccounting** ……下線(会計)<br>**xlDoubleAccounting** ……二重下線(会計) |
| | p | **Font.Subscript** | gs | | **True** ＝ 下付き文字 |
| | p | **Font.Superscript** | gs | | **True** ＝ 上付き文字 |
| 名前 | p | **Font.Name** | gs | | フォントオブジェクトの名前 |
| 親 | p | **Font.Application** | g | | フォントを作成したアプリケーションを返す |
| | p | **Characters.Application** | | | キャラクタを作成したアプリケーションを返す |
| | p | **Font.Creator** | g | M | フォントを作成したオブジェクトを32ビットの整数型(4文字のシグネチャ)で返す |
| | p | **Characters.Creator** | g | M | 文字列を作成したオブジェクトを32ビットの整数型(4文字のシグネチャ)で返す |
| | p | **Font.Parent** | g | | 親オブジェクト |

(注) m：メソッド，p：プロパティ，M：メソッドあり，g：取得可能，gs：取得・設定とも可能

〈表9-3〉 Caption プロパティの対象となる主なオブジェクトの機能

| オブジェクトの種類 | 機　能 |
|---|---|
| **Application** | Microsoft Excel のメインウィンドウのタイトルバーに表示される名前を指定．名前を省略するか，**Empty** に設定すると，このプロパティの値は“**Microsoft Excel**”となる．Macintosh では，値の取得のみ． |
| **AxisTitle** | 軸タイトルの文字列 |
| **Button** | ボタンの文字列 |
| **Characters** | 当該範囲の文字列 |
| **ChartTitle** | グラフタイトルの文字列 |
| **Controls** | コントロール(チェックボックス，ダイアログシート上のフォーム，ドロップダウン，エディットボックス，グループボックス，ラベル，オプションボタン)の文字列 |
| **DataLabel** | データラベルの文字列 |
| **Menu** | メニューの名前．アクセスキーにしたい文字があるときは，その文字の直前にアンパサンド(**&**)を置く．アルファベット・アクセスキーとカタカナ・アクセスキーを指定するには，“**ファイル(&F&フ)**”と記述．［コントロールパネル］の［キーボードの設定］でアクセスキーがアルファベットに設定されているときは，“**ファイル(F)**”となり，カタカナに設定されているときは，“**ファイル(フ)**”と表示する． |
| **MenuBar** | メニューバーの文字列． |
| **MenuItem** | メニュー項目の名前．アクセスキーにしたい文字があるときは，その文字の直前にアンパサンド(**&**)を置く．アルファベット・アクセスキーとカタカナ・アクセスキーを指定するには，“**終了(&X&リ)**”と記述．［コントロールパネル］の［キーボードの設定］でアクセスキーがアルファベットに設定されているときは，“**終了(X)**”となり，カタカナに設定されているときは，“**終了(リ)**”と表示する． |
| **TextBox** | テキスト ボックス内のテキスト |
| **Window** | ドキュメントウィンドウのタイトルバーに表示される名前．名前を指定しておくと，その名前を **Windows** プロパティに対するインデックスとして使用できる．既定のドキュメント名に戻したいときは，**Caption** プロパティを **Empty** に設定する． |

〈表9-4〉 文字列操作関数(ワークシート)一覧

| 関数 | 機能 |
|---|---|
| 検索 | |
| **FIND** | 指定した文字列を別の文字列の中で検索．大文字と小文字は区別． |
| **FINDB** | 指定した文字列を別の文字列の中で検索(バイト単位)．大文字と小文字は区別． |
| **SEARCH** | 指定した文字列を別の文字列の中で検索．大文字と小文字は区別しない． |
| **SEARCHB** | 指定した文字列を別の文字列の中で検索(バイト単位)．大文字と小文字は区別しない． |
| 変換(全角，半角) | |
| **ASC** | 文字列内の全角の英数カナ文字を，半角文字に変換． |
| **JIS** | 文字列内の半角の英数カナ文字を，全角文字に変換． |
| 変換(数値→文字) | |
| **TEXT** | 数値を書式設定した文字列に変換． |
| **DOLLAR** | 数値を四捨五入し，ドル書式を設定した文字列に変換． |
| **FIXED** | 数値を四捨五入し，書式設定した文字列に変換． |
| **YEN** | 数値を四捨五入し，円書式を設定した文字列に変換． |
| 変換(文字→数値) | |
| **VALUE** | 文字列を数値に変換． |
| 変換(大文字，小文字) | |
| **LOWER** | 文字列に含まれる英字をすべて小文字に変換． |
| **UPPER** | 文字列に含まれる英字をすべて大文字に変換． |
| **PROPER** | 文字列に含まれる英単語の頭文字だけを大文字に変換． |
| 変換(ASCII，文字) | |
| **CHAR** | 数値を ASCII または JIS コード番号と見なし，それに対応する文字を返す． |
| **CODE** | 文字列の先頭文字に対応する ASCII または JIS コードを返す． |
| 長さ | |
| **LEN** | 文字列に含まれる文字数を返す． |
| **LENB** | 文字列に含まれるバイト数を返す． |
| 部分文字列 | |
| **LEFT** | 文字列の先頭(左端)から指定した数の文字を返す． |
| **LEFTB** | 文字列の先頭(左端)から指定したバイト数の文字を返す． |
| **RIGHT** | 文字列の末尾(右端)から指定した数の文字を返す． |
| **RIGHTB** | 文字列の末尾(右端)から指定したバイト数の文字を返す． |
| **MID** | 文字列の任意の位置から指定した数の文字を返す． |
| **MIDB** | 文字列の任意の位置から指定したバイト数の文字を返す． |
| 置換 | |
| **SUBSTITUTE** | 文字列中の指定した文字を別の文字に置き換える． |
| **REPLACE** | 文字列中の指定した数の文字を別の文字に置き換える． |
| **REPLACEB** | 文字列中の指定したバイト数の文字を別の文字に置き換える． |
| その他 | |
| **CLEAN** | 文字列から印刷できない文字を削除． |
| **CONCATENATE** | 複数の文字列を結合して一つの文字列にまとめる． |
| **EXACT** | 二つの文字列が等しいかどうかをテスト． |
| **REPT** | 文字列を指定した回数だけ繰り返して表示． |
| **T** | 引数を文字列に変換． |
| **TRIM** | 文字列から余分なスペースを削除． |

〈表9-5〉 Basic の文字列処理関数一覧

■ データの変換

| 関数 | 説明 |
|---|---|
| 文字列を ASCII に | |
| **Asc( )** | 文字列の最初の文字を ASCII コードに変換 |
| ASCII を文字列に | |
| **Chr( )** | 指定した文字コードに対応する文字を返す |
| 数値を文字列に | |
| **Format( )**<br>**Str( )** | 文字列を指定した書式にしたがって変換する<br>数式の値を文字列で表した値で返す |
| 文字列を数値に | |
| **Val( )** | 文字列で表現した数字を，数値に変換 |
| 8 進/16 進を文字列に | |
| **Hex( )**<br>**Oct( )** | 数値を 16 進数で表した文字列に変換<br>引数に指定した値を 8 進数で表した文字列に変換 |

■ 文字列の処理

| 関数 | 説明 |
|---|---|
| 比較モード | |
| **Option Compare** | 文字列データの比較の際，比較モードを宣言する |
| 文字コード変換 | |
| **Asc( )**<br>**Chr( )** | 文字列の最初の文字の文字コードを返す<br>指定した文字コードに対応する文字を返す |
| 文字列の比較 | |
| **StrComp( )** | 文字列比較の結果を示す値を返す |
| 文字列の検索 | |
| **InStr( )** | 指定した文字列が，ある文字列の何文字目にあるかを取得する |
| 大文字と小文字 | |
| **Format( )**<br>**LCase( )**<br>**UCase( )** | 文字列を指定した書式にしたがって変換する<br>アルファベットの大文字を小文字に変換する<br>文字列中の小文字を大文字に変換する |
| 文字の繰り返し | |
| **Space( )**<br>**String( )** | 指定した数のスペースからなる文字列を返す<br>指定した文字コード(ASCII またはシフト JIS コード)の示す文字，または文字列の先頭文字を，指定した文字数だけ並べた文字列を返す |
| 空白の削除 | |
| **Ltrim( )**<br>**Rtrim( )**<br>**Trim( )** | 指定した文字列の先頭からのスペースを削除した文字列を返す<br>指定した文字列の末尾からスペースを削除した文字列を返す<br>指定した文字列の先頭と末尾の両方からスペースを削除した文字列を返す |
| 文字列の長さ | |
| **Len( )** | 指定した文字列の文字数，変数に必要なバイト数を返す |
| 文字列の書式 | |
| **Format( )** | 文字列を指定した書式にしたがって変換する |
| 文字列の配置 | |
| **Lset**<br>**Rset** | 文字列変数の値を左詰めにしたり，ユーザー定義型の変数を別のユーザー定義型の変数にコピーする<br>文字列変数の値を右詰めにする |
| 文字列検索 | |
| **InStr( )** | ある文字列の中から別の文字列を検索し，最初に見つかった文字位置を返す |
| 文字列置き換え | |
| **Mid( )** | 文字列変数の一部を他の文字列に置き換える |
| 文字列取り出し | |
| **Left( )**<br>**Right( )** | 文字列の左端から指定した文字数分の文字列を取り出す<br>文字列の右端から指定した文字数分の文字列を取り出す |

の指定ができます．

```
With Range("A1").Font
  .FontStyle = "MS 明朝"
  .Size = 16
  .Italic = True
End With
Range("A1").Characters(1, 1).Font.Bold = True
```

とすると，最初の文字だけを太字にできます(図9-5)．

# VBAプログラミングアドバイス

## OLEオートメーションとは

マイクロソフト社の発表によれば，8月24日に米国で，英語版Windows95の発売が開始されることが確実になりました．これに合わせて，Microsoft OFFICEもOFFICE for Windows95と，新しいバージョンになるものと思われます．

また，今年中に発表されると思われる英語版Accessの新バージョンにもVBAが搭載されることが確実です．そのほかにも，秋頃にはProjectの日本語版を使うことができそうですが，これにもVBAが搭載されているようですし，SQL 6.0にもVBAは搭載されているようです．

何よりも期待の大きいVisualBasic 4.0は，基本的にはVBAのエンジンです．このように，マイクロソフト社のアプリケーションのすべてにVBAが搭載されるようになるのも，そう遠い話ではないでしょう．

ここで，大きな意味を持ってくるのがOLEオートメーションです．OLEオートメーションの世界では，アプリケーションの内部が完全にクラスになっており，オブジェクト，メソッド，プロパティなどをアプリケーションの外部から操作できるようになります．

### ● ExcelからWordを動かす

そこで今回は，このOLEオートメーションを実験してみることにします．VBAには，**CreateObject**という関数がありますが，この関数を使うことによって，OLEオートメーションオブジェクトを生成することができます．

**CreateObject関数の書式**

**CreateObject(appname.objecttype)**

**appname** …OLEオートメーションオブジェクトを提供しているアプリケーション名

**objecttype** …作成するオブジェクトの種類またはクラス

Excelは OLEオートメーションのサーバーにもクライアントにもなれますし，Wordはサーバーになれます．そこで，ExcelのVBAからWordを使ってみましょう．

オブジェクトですから，オブジェクト型の変数を宣言しておく必要があります．

```
Dim wdBasic As Object
```

Wordは，残念ながら**Basic**というオブジェクトしか提供してくれません．そこで，

```
Set wdBasic = CreateObject("Word.Basic")
```

で，オブジェクトを宣言します．

こうしておけば，あとはWord Basicのコマンドを VBAの上で自由に使うことができます．次に，簡単なプログラムを書いてみましょう．

```
Sub ExceltoWord( )
  Dim wdBasic As Object
  Set wdBasic = CreateObject("Word.Basic")
  With wdBasic
       .FileNew
       .InsertPara
       .Insert "初めてのOLEオートメーション"
       .InsertPara
       .Insert "記念すべき日は," & Date$( ) & "です"
       .FileSaveAs "e:¥oletest.doc"
   End With
   Set wdBasic = Nothing
   End
End Sub
```

このプログラムは，Wordを起動して，文書に1行空けて「初めてのOLEオートメーション」と書き込み，さらに1行空けて「記念すべき日は，1995-07-05です」と書き込み，**oletest.doc**という名前でセーブする，単純なものです．

Word Basicをご存じの方でしたら，このプログラムのほとんどが，Word Basicのステートメントに**wdBasic.**を書いただけのものであることがお分かりになると思います．すなわち，Word BasicのコードをExcelの上で書くことができるのです．

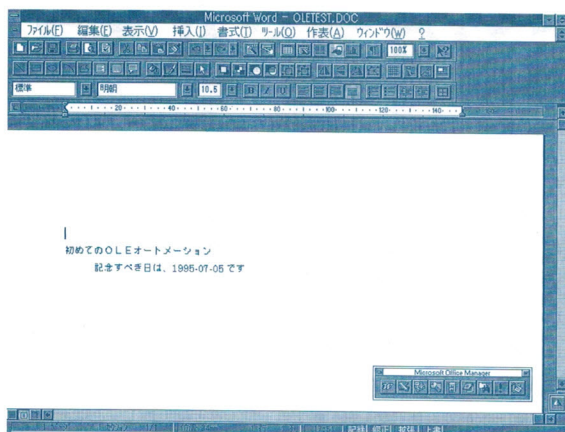
?!

通常，Wordを起動すると，「文書1」のようなファイルを自動的にオープンしますが，OLEオブジェクトとして呼び出した場合には，データファイルはオープンしません．ですから，**FileNew**が必要なのです．これは，Excelなどでも同じです．

実際にExcelから起動すると，Wordが動いているのを画面で見ることができます．もし画面に現れるのがいやでしたら，**.FileNew**の前に**.AppMinimize**と書けば，Wordは見えません．

この**oletest.doc**をWordで開いてみれば，実際に書き込みが行われているのを確認できます(図9-A)．

**Set wdBasic = Nothing**は生成したオブジェクトを解放するもので，Wordは**Set wdBasic = Create**

〈図9-A〉 Excel から Word Basic を起動する

**Object("Word.Basic")** で起動し，このコマンドで終了します．

Word の場合は，呼び出せるオブジェクトは一つでしたが，Excel の場合は，**Worksheet**，**Chart** など，いろいろなオブジェクトも直接呼び出すことができます．

VBA のコマンドはそのまま使えますから，とても便利です．ただ残念なことに，OLE オートメーションのクライアントをサポートしているソフトが日本にはありません．

そこで，VisualBasic 3.0 の英語版で，Excel を利用する例を示しましょう．VBA で書く場合と書式はほとんど同じですが，VisualBasic 3.0 では **With** が使えませんから，少し工夫しています．

```
Sub testOLE( )
    Dim excelObj As Object
    Dim rangeObj As Object

    Set excelObj = CreateObject
                        ("Excel.Application")

    excelObj.Workbooks.Open
            "d:¥msoffice¥excel¥sample¥sales.xls"

    excelObj.Sheets(1).Select
    Set rangeObj = excelObj.Range("B4:D4")
    rangeObj.HorizontalAlignment = xlCenter
    rangeObj.VerticalAlignment = xlBottom
    rangeObj.WrapText = False
    rangeObj.Orientation = xlHorizontal
    rangeObj.AddIndent = False

    Dim fontObj As Object
    Set fontObj = rangeObje.Font
    rangeObj.Name = "MS 明朝"
    rangeObj.FontStyle = "太字"
    rangeObj.Size = 12
    Rows("B:D").Autofit
    Set excelObj = Nothing
    End
End Sub
```

これは，Excel で提供されている **sales.xls** のタイトル部分を少し整形するプログラムです．

Excel は，**CreateObject** 関数で起動しても，画面には見えません．もし画面でみたい場合には，オブジェクトを呼び出した際に，次の1行を追加してください．

```
excelObject.Visible = True
```

### ● ファイルからオブジェクトを呼び出す

また，OLE オートメーションオブジェクトをデータファイルから取得する関数 **GetObject( )** もあります．

**GetObject( ) の書式**

**GetObject("ファイル名"[, "アプリケーション名.オブジェクトの種類"])**

**ファイル名**……取得したいオブジェクトが含まれているファイルの絶対パスおよびファイル名を指定．ここで，空の文字列(" ")を指定した場合には，**"アプリケーション名.オブジェクトの種類"** を指定しなければなりません．

**"アプリケーション名.オブジェクトの種類"**
……**CreateObject( )** と同じ

**CreateObject( )** が，アプリケーションのオブジェクトを取得したのに対して，こちらはデータファイルを呼び出して，指定したファイル名に対応するアプリケーションを起動し，指定したファイル内のオブジェクトをアクティブにし，そのオブジェクトを取得します．

通常は，データファイルは自分を作成したアプリケーションを知っていますから，アプリケーション名を指定する必要はありません．さきほどの場合でも，

```
Set excelObj =
GetObject("d:¥msoffice¥excel¥sample¥sales.xls")
```

では，Excel が起動して，ワークブック **sales.xls** を読み込み，オブジェクトとして参照できるようにします．

また，

```
Set excelObj = GetObject("e:¥data¥uriage.
xls", "Application.Sheets("売上日報")")
```

なら，**e:¥data¥uriage.xls** の中の「売上日報」というワークシートをオブジェクトとして参照できるようにします．

### ● Word を呼び出す場合の注意

Word の OLE は，少しおかしなところがありますので，注意点を挙げておきます．

第一に，オブジェクト変数がスコープを失った場合や，**Nothing** を代入して，オブジェクトへの参照関係を解放した場合には，Word が終了してしまうのです．

ですから，WordBasic を OLE オブジェクトとして参照する際には，オブジェクト変数をグローバル変数として宣言しなければいけません．

```
Gloval wdBasic As Object
Set wdBasic = CreateObject("Word.Basic")
```

# Windows 上で「A:¥>dir | sort > test.txt」を行うには

山本　孝

## Q. Windows 上で「A:¥>dir | sort > test.txt」を行うには

**もしかすると，どこかで親切に解説されていることなのかもしれませんが，次のような疑問にぶち当たっています．**

**DOS で日常的にやっていた，**

**A:¥>DIR | sort > test.txt**

**を Windows のプログラムの中で行うにはどうしたらいいのでしょうか．**

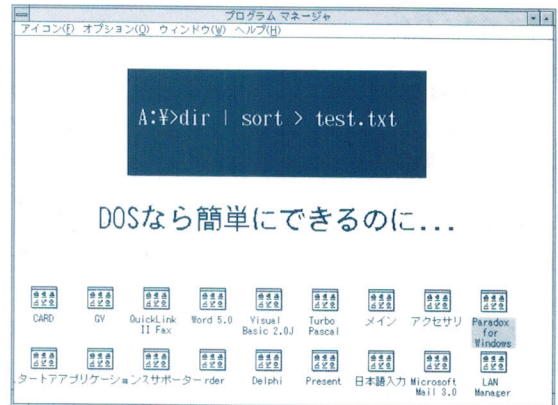
**「指定ディレクトリのファイル名，サイズ，日付けの一覧をテキストファイルに落とす」ことがなかなかできなくて困っています．**

**もちろん DOSBOX を使えば，パイプはともかくとして何とかなります．また，リストボックスを使えばファイル名は取得できそうですが，ひどく煩雑で遠回りをしているように思います．何か簡単なコマンドが用意されているはずだ，と信じているのですが見つかりません．**

**（杉本忠雄さんからの質問）**

## A. _dos_find 系/FindFirstFile 系の関数を使う

「Windows なんだから，こんなことは一発で簡単にできる何かがあるんだろう」とお思いでしょうが，実際は，一発で簡単にできる方法はありません．…残念でした．

〈図5-1〉 Windows 3.1 で，| や > くは難しい

ただし，ファイル名の一覧を出すだけならば，**DlgDirList( )** という API があります．

しかしこれは，ダイアログボックスのために用意されている API のため，いちいちリストボックスを作成しなければならず，日付けを知ることもできません．

そこで，目的を達成するために三つの案を思い付きました．

### ● _dos_find 系の関数を使う

ランタイムライブラリに，**_dos_findfirst( )** と **_dos_findnext( )** という関数があります．

これは，目的のファイルを見つけ出し，列挙するという関数です．この関数は，ファイルを列挙すると同時に，タイムスタンプなども **_find_t** 構造体に返してくれるので，この値などを変換してファイルに書き出せ

〈リスト5-1(a)〉 _dos_find 系の関数を使ったサンプルプログラム

```
#include <windows.h>
#include <dos.h>

HFILE  hfile;

void write( struct _find_t * pfind ) {
  char buff[1024];
  wsprintf( (LPSTR)buff,
            "%13s  %4d年%2d月%2d日  %2d時%2d分¥t",
      (LPSTR)pfind->name,
      1980 + ( ( pfind->wr_date & 0xFE00 ) >> 9 ),
      ( pfind->wr_date & 0x01E0 ) >> 5,
      ( pfind->wr_date & 0x001F ) >> 0,
      ( pfind->wr_time & 0xF800 ) >> 11,
      ( pfind->wr_time & 0x07E0 ) >> 5
      );
  lstrcat( (LPSTR)buff,
      ( pfind->attrib & _A_ARCH ) ? "A" : " " );
  lstrcat( (LPSTR)buff,
      ( pfind->attrib & _A_HIDDEN ) ? "H" : " " );
  lstrcat( (LPSTR)buff,
      ( pfind->attrib & _A_RDONLY ) ? "R" : " " );
  lstrcat( (LPSTR)buff,
      ( pfind->attrib & _A_SYSTEM ) ? "S¥n" : " ¥n" );
  _lwrite( hfile, (LPSTR)buff,
                  lstrlen((LPSTR)buff) );
}

int PASCAL WinMain( HANDLE, HANDLE, LPSTR, int ) {
  struct _find_t  find;
  hfile = _lcreat( "c:¥¥test.txt", 0 );
  if( _dos_findfirst( "c:¥¥*.*",
          _A_NORMAL | _A_ARCH, &find ) == 0 ) {
    do {
      write( &find );
    } while( _dos_findnext( &find ) == 0 );
  }
  _lclose( hfile );
  return 0;
}
```

ばよいわけです.

ただし，32ビットアプリケーションでは，**_dos_find**系の関数は使えないはずです．その場合は，代わりに新しく用意されたAPIである**FindFirstFile( )**と**FindNextFile( )**を使うとよいでしょう．

**リスト5-1**は，**_dos_find**系を使ったサンプルプログラムです．しかしぜんぜんWindowsとは関係のないプログラムだな….

### ● やっぱりDlgDirList( )を使う

**_dos_find**系で列挙するのは面倒なので，やはり**DlgDirList( )**(もしくは**LB_DIR**メッセージ)を使ってファイル一覧を作ることも可能です．

**DlgDirList( )**は，リストボックスが必要なので，ダイアログボックスにリストボックスを付けて呼び出すか，**CreateWindow( )**でクラス名に“**ListBox**”を指定して自分で作らなくてはいけません．

リストボックスが作成できたら，**DlgDirList( )**を呼んでリストボックスにファイル名一覧を入れます．

ファイル名一覧ができたら，リストボックスから1項目ごとファイル名を取り出して，**_dos_getftime( )**を使えばファイルのタイムスタンプがわかるので，変換してファイルに書き出せばでき上がりです．

Win32の場合,**GetFileTime( )**というAPIがあるのでこれを使うとよいでしょう．

しかしやはり面倒ですね．

### ● パイプを使う

パイプといえば,DOSのアプリケーションのみ使えるとお思いでしょうが，実は32ビットのWindowsアプリケーションでも使うことができます．

つまり，DOSで日常的にやっていた

**A:¥>DIR | sort > test.txt**

ということがWindowsプログラムでも可能なのです．

なんだ，やっぱりできるんじゃないか，と思いきや，これをやるための手順がかなり面倒なんです．

32ビットアプリケーションは，DOSと同じように標準入力と標準出力をもっており，リダイレクトすることができます．**リスト5-2**，**リスト5-3**は，標準入出力をリダイレクトしてプログラムを呼び出すサンプルプログラムです．

**CreatePipe( )**で，名前なしパイプと呼ばれる通信路を作成し，**CreateProcess( )**を呼ぶパラメータにパイプのハンドルを渡します．

**CreatePipe( )**は二つのハンドルを返し，一方はデー

〈リスト5-1(b)〉 リスト5-1(a)の実行結果

```
 COMMAND.COM  1994年 4月15日   6時30分 A
$JPNHN16.FNT  1994年 4月15日   6時30分 A
$JPNHN19.FNT  1994年 4月15日   6時30分 A
$JPNZN16.FNT  1994年 4月15日   6時30分 A
$SYS1DIC.FNT  1994年 4月15日   6時30分 A
  WINA20.386  1994年 4月15日   6時30分 A
$JPNHN24.FNT  1994年 4月15日   6時30分 A
$JPNZN24.FNT  1994年 4月15日   6時30分 A
$JPNZN12.FNT  1994年 4月15日   6時30分 A
$JPNZN24.IBM  1994年 4月15日   6時30分 A
$JPNHN12.FNT  1994年 4月15日   6時30分 A
$SYS1Z16.FNT  1994年 4月15日   6時30分 A
$SYS1Z24.FNT  1994年 4月15日   6時30分 A
AUTOEXEC.OLD  1995年 2月26日  21時 7分 A
  CONFIG.PS2  1995年 2月26日  21時 7分 A
AUTOEXEC.PS2  1995年 2月26日  21時 7分 A
  CONFIG.OLD  1995年 2月26日  21時 7分 A
  CONFIG.SYS  1995年 6月29日  21時51分 A
AUTOEXEC.BAT  1995年 6月29日  21時52分 A
  CONFIG.SYD  1995年 4月 5日  10時37分 A
BOOTSECT.DOS  1995年 2月26日  22時44分 A
PAGEFILE.SYS  1995年 7月 7日  14時14分 A
  CONFIG.BAK  1995年 4月 5日  12時31分 A
     AAA.TXT  1995年 4月26日  18時42分 A
  CONFIG.001  1995年 4月 5日  12時37分 A
AUTOEXEC.001  1995年 4月20日  19時52分 A
AUTOEXEC.BAK  1995年 6月29日  21時51分 A
     AAA.BAK  1995年 4月26日  18時34分 A
     DDD.TXT  1995年 5月30日  17時29分 A
HIDEMARU.KEY  1995年 6月 2日  14時52分 A
    TEST.TXT  1995年 7月 7日  15時34分 A
```

〈リスト5-2〉 パイプを使ったサンプルプログラム (Win32用)

```
#include <windows.h>

HANDLE hForwardDoor;
HANDLE hForwardExit;
HANDLE hBackwardDoor;
HANDLE hBackwardExit;

////////////////////////////////////////////////////////////
/
// Connect()
// 標準入出力をリダイレクトしてプログラムを起動する関数
////////////////////////////////////////////////////////////
/
BOOL Connect( LPSTR pszExe ) {
  SECURITY_ATTRIBUTES sa;
  sa.nLength = sizeof(sa);
  sa.bInheritHandle = TRUE;
  sa.lpSecurityDescriptor = NULL;

  //標準出力をリダイレクトするためのパイプ
  if( !CreatePipe( &hBackwardExit,
           &hBackwardDoor, &sa, 0 ) ) return FALSE;

  //標準入力をリダイレクトするためのパイプ
  if( !CreatePipe( &hForwardExit, &hForwardDoor,
                  &sa, 0 ) ) return FALSE;
  HANDLE hDummy = hForwardDoor;
  DuplicateHandle( GetCurrentProcess(), hDummy,
        GetCurrentProcess(), &hForwardDoor, 0,
        FALSE, DUPLICATE_SAME_ACCESS );
  CloseHandle( hDummy );

  //子プロセスの起動
  PROCESS_INFORMATION  pi;
  STARTUPINFO si;
  memset( &si, 0, sizeof(si) );
  si.cb = sizeof(si);
  si.dwFlags = STARTF_USESTDHANDLES;
  si.hStdInput = hForwardExit;   //標準入力のリダイレクト
  si.hStdOutput = hBackwardDoor; //標準出力のリダイレクト
  if( !CreateProcess(NULL, pszExe, NULL, NULL, TRUE,
          DETACHED_PROCESS, NULL, NULL, &si,
                                     &pi ) ) {
    MessageBox( NULL, "実行失敗", "", MB_OK );
    return FALSE;
  }
  if( pi.hProcess ) CloseHandle( pi.hProcess );
  if( pi.hThread ) CloseHandle( pi.hThread );

  return TRUE;
}
```

タを送る側，もう一方はデータを受け取る側のハンドルになります．

さらに，1本のパイプでは，1方向の通信しかできないので，標準入力用，標準出力用と2本のパイプを作成しなくてはいけません．計四つのハンドルを扱うことになります．

次に，例えば，「**dir ｜ sort ＞ test.txt**」を行いたい場合，2段階に分けてプログラムを呼び出します．これは「**dir**」を呼び出すためと，「**sort**」と呼び出すためです．二つのハンドルのパイプを2本作って，さらに2段階…ややこしい．

最初の「**dir**」の部分ですが，**dir** は **EXE** ファイルではないので，**CreateProcess( )** で呼び出すことができませんが，**cmd.exe** を経由して呼び出すことができます．

**"cmd.exe /c dir"** というコマンドラインで **CreateProcess( )** を呼び出せば，**dir** を実行することができます．

**dir** の結果(つまり標準出力の結果)は，パイプを通して **ReadFile( )** で読み出します．読み出したデータは次に **sort.exe** に渡さなくてはいけません．

「**| sort**」の部分ですが，これは **sort.exe** というファイルがあるので **CreateProcess( )** で呼び出せます．

呼び出したら，先程取り出した「**dir** の結果」を，パイプを通して **WriteFile( )** することで，**sort** の標準入力に「**dir** の結果」を渡せます．

渡し終わったら，またパイプを通して **ReadFile( )** でソート結果を取り出せます．

そして，そのソート結果を **test.txt** に書き出せば，「**dir ｜ sort ＞ test.txt**」と同等のことができるわけです．

ああ，面倒．しかも本来の目的を完遂できていない．

ちなみに，Windows 3.1のアプリケーションではこの手は使えません．あんまりいいことないですね．

### ● 結論

結論として，最初の **_dos_find** 系関数，または **FindFirstFile( )** 系の関数を使ったほうがよいと思います．

〈リスト5-2〉パイプを使ったサンプルプログラム(Win32用，つづき)

```
////////////////////////////////
// Transmit()
// 標準入力にデータを渡す関数
////////////////////////////////

BOOL Transmit( HGLOBAL hMem ) {
  LPSTR  lpMem;
  BOOL fRes;
  DWORD  cbWritten;
  lpMem = (LPSTR)GlobalLock( hMem );
  fRes = WriteFile( hForwardDoor, lpMem,
              lstrlen(lpMem), &cbWritten, NULL );
  if( fRes == FALSE || cbWritten == 0 ) return FALSE;
  GlobalUnlock( hMem );
  CloseHandle( hForwardDoor );
  return TRUE;
}

/////////////////////////////////
// Recieve()
// 標準出力からデータもらう関数
/////////////////////////////////
HGLOBAL Recieve( void ) {
  HANDLE hMem;
  LPSTR  lpMem;
  DWORD  cbMem = 8192;
  DWORD  cbSize = 0;
  CloseHandle( hBackwardDoor );
  hMem = GlobalAlloc( GMEM_MOVEABLE, cbMem );
  lpMem = (LPSTR)GlobalLock( hMem );
  for(;;) {
    DWORD  cbRead;
    BOOL fRes;
    fRes = ReadFile( hBackwardExit, lpMem + cbSize,
            cbMem - cbSize, &cbRead, NULL);
    if( fRes == FALSE || cbRead == 0 ) break;
    cbSize += cbRead;
    if( cbMem - cbSize < 8192 ) {
      cbMem += 8192;
      GlobalUnlock( hMem );
      hMem = GlobalReAlloc( hMem, cbMem,
                          GMEM_MOVEABLE );
      lpMem = (LPSTR)GlobalLock( hMem );
    }
  }
  lpMem[cbSize] = '¥0';
  GlobalUnlock( hMem );
  CloseHandle( hBackwardExit );
  return hMem;
}
```

## Q. スクリーンセーバーをプログラムから停止する方法は

**スクリーンセーバーを起動させる方法はわかるのですが，プログラムから停止させる方法がわかりません．**

〈リスト5-3〉リスト5-2を使用して，パイプやリダイレクトを実行するサンプルプログラム(Win32用)

```
//////////////////////////////////////////
// WinMain()
// 「dir c:¥ | sort >c:¥test.txt」 をやる
//////////////////////////////////////////
int PASCAL WinMain( HANDLE, HANDLE, LPSTR, int ) {
  HGLOBAL   hMem;

  // dir の部分
  Connect( "cmd /c dir c:¥¥" );
  hMem = Recieve();

  // | sort の部分
  Connect( "sort" );
  Transmit( hMem );
  GlobalFree( hMem );
  hMem = Recieve();

  // >test.txt の部分
  LPSTR  lpMem;
  HFILE  hfile;
  lpMem = (LPSTR)GlobalLock( hMem );
  hfile = _lcreat( "c:¥¥test.txt", 0 );
  _lwrite( hfile, lpMem, lstrlen(lpMem) );
  _lclose( hfile );
  GlobalUnlock( hMem );
  GlobalFree( hMem );

  return 0;
}
```

## A. WM_SYSCOMMAND メッセージを自分で処理する

スクリーンセーバーが起動する前に，Windowsはアクティブなアプリケーションに対して，**SC_SCREENSAVE**を指定した**WM_SYSCOMMAND**メッセージを送ります．

このメッセージは，「スクリーンセーバーを起動する」というメッセージです．ですから，自分で自分自身にこのメッセージを送ってやれば，自発的にスクリーンセーバーを起動することができます(**リスト5-4**)．

逆に，このメッセージを**DefWindowProc( )**に渡さずに，自分で処理すれば，スクリーンセーバーを抑制することができます(**リスト5-5**)．

このときは，戻り値に0以外の値を渡さなくてはいけません．ちなみに，**WM_SYSCOMMAND**はアクティブなアプリケーションにしか送られてこないので，自分がアクティブなときにしか抑制できません．

〈リスト5-4〉スクリーンセーバーを自分で実行する

```
SendMessage( hwnd, WM_SYSCOMMAND, SC_SCREENSAVE, 0 );
```

〈リスト5-5〉スクリーンセーバーを抑制する

```
switch( message ) {
  case WM_SYSCOMMAND:
    if( wParam == SC_SCREENSAVE ) {
      return 1;
    }
    break;
}
```

今回は，この二つで質問はおしまいです．

二つじゃ寂しいので，みんなからのお便りドシドシまってるよぉー(ホント)．

＊　＊　＊

このコーナーでは，読者の皆様からの質問を受け付けています．お手紙でも，NIFTY-ServeのIDへのメール(HBG01366)でも何でもOKです．

# パソコン通信による掲載プログラムの入手方法

本誌に掲載したプログラムは，フロッピ・ディスクで入手できますが（159ページ参照），そのほかにパソコン通信でも入手することができます．

パソコン通信で入手するためには下記のものが必要です．

パソコン，モデム，通信ソフト（Windowsのアクセサリに含まれているターミナルでも可能です）．

### ●登録場所

NIFTY-Serveの「パソコン工作フォーラム」のデータライブラリの中に「CQ出版トランジスタ技術提供データ」があります（図1）．登録期間は次号が発売されるまでの約1ヵ月間で，毎月18日更新を目安にしますが，数日前後することもあります．

### ●頒布ディスクとの違い

NIFTY-Serveのライブラリに登録するデータは，頒布ディスクのものと若干内容が異なります．ライブラリ登録版はソース・プログラムだけですが，頒布ディスク版ではコンパイル済みの実行ファイルが添付されているものもあります．

データをダウンロードするためにはNIFTY-Serveへの入会が必要です．そのための詳細は下記へお問い合わせください．

【問い合わせ先】ニフティ(株)メンバーサービス部
☎03(5471)5806

〈図1〉トップ・メニュからの操作手順

```
NIFTY-Serve     TOP
 1. サービス案内・検索              2. 電子メール
 3. 掲示板                          4. CBシミュレーター
 5. フォーラム                      6. ニュース／スポーツ／天気予報
 7. 企業／経済／人物／行政          8. 科学／技術／翻訳
 9. 書籍／教育／就職               10. 趣味／旅行／生活
11. 占い／クイズ／ゲーム           12. ワープロ／コンピューター
13. ショッピング／ステーション     14. 海外データベース(INFOCUE)
15. コンピュサーブコーナー         16. ビギナーズコーナー
17. 他ネット接続サービス            E. 終了
>go fdevice

ｘｘ　ｘｘさん　花火、絵日記、スイカ割り・・・
パソコン工作フォーラムへようこそ。

<パソコン工作フォーラム>    FDEVICE
1:お知らせ           2:掲示板      3:電子会議
4:データライブラリ   5:会員情報    6:リアルタイム会議
7:SYSOP 宛メール     8:オプション  E:終了
>4
番号  総数  登録済   最新   ライブラリ名
  1    66  (  66)  06/19  HDD,FDD,HDCの仕様集
  2    92  (  92)  07/09  BIOS,ﾄﾞﾗｲﾊﾞ-関連
  3    89  (  88)  07/17  とりあえずデータ．なんでも
  4   216  ( 215)  07/24  アプリケーション・ソフト
  5    52  (  52)  12/07  回路図
  6    71  (  71)  07/07  満員御礼、閉鎖会議室の全文保管場所
  7    71  (  37)  07/18  CQ出版トランジスタ技術提供データ
  8    54  (  54)  06/28  試作・公開期間限定データ＆ソフト
  9    37  (  37)  03/27  旧バージョン・データ&ソフト
>7
データライブラリ (1:データ一覧  2:検索  3:ｱｯﾌﾟﾛｰﾄﾞ (無料)   4:ﾀﾞｳﾝﾛｰﾄﾞ  E:終了)
>1
番号  ID        登録日付 バイト  参照   データ名
 78  HBD01761 95/07/17  30552  398 B トラ技コンピュータ1995年8月号
 69  HBD01763 94/12/07  12956  772 B ﾄﾗﾝｼﾞｽﾀ技術,年間総目次,1994,TR94ONLY.LZH
 67  HBG01366 94/10/18  78156 1666 B パソコン通信プログラムのすべて
 64  HBG01366 94/09/13  41118  611 B トラ技コンピュータデータベース
 58  HBG01366 94/07/13  31013 1047 B トラ技コンピュータ1994年7月号特集訂正

>78
```

データの概要が表示され，その後実際のダウンロードを行う．ダウンロードの操作手順は通信ソフトにより異なるのでここでは省略する．

# CD-ROMレポート⑧

《WINDOWS MANIA, DR.REFERENCE, WIN EXPERT CD-ROM》

＊本稿で紹介したCD-ROMを1セットずつ読者に抽選でプレゼントします．希望される方は，アンケートはがきでご応募ください．

問い合わせ先▶パシフィック・ハイテック・ニューキッズ
〒101　東京都千代田区外神田1-8-10
☎03-3255-6855

今回は，Windowsプログラマに役立つソフトウェアを紹介します．WINDOWS MANIAは，その名のとおりごく一部のユーザーの間でのみ知られたタイトル(図1)．バージョンは古いのですが，是非手に入れたい一枚です．

Dr.REFERENCEは，お馴染みACES RESEARCH社の定番で，こちらも息の長いタイトル，そして，WIN EXPERT CD-ROMです(図2，図3)．

▶〈図1〉WINDOWS MANIA

◀〈図2〉DR.REFERENCE

▶〈図3〉WIN EXPERT CD-ROM

● Electrified Visual Basic Magazine(WINDOWS MANIA)

ヘルプファイル形式で配布されている，Visual Basic(VB)に関するトピックを集めたファイル．

商用向け，またはシェアウェアなどで作成されているVB関係の製品に関するレビューやプログラミング教室，tipsなどが収録されている．

「ヘルプファイル形式で配布する雑誌というのはなかなか目新しい．」というのが作者の弁．

| ソフト名 | **Electrified Visual Basic Magazine V.3.10.160** |
|---|---|
| 作者(発行者) | Don Welsh |
| 分類/登録料 | Shareware(配布に関する制限無し)/$4.50 |
| 動作環境 | Windows 3.1 |
| 動作確認 | NEC PC-9821Ce+MS-DOS 6.2+NEC製MS-Windows 3.1 |
| 所在 | **WINDOWS MANIA CD-ROM ¥PROGRAM ¥EVBMAG.ZIP** |

● Programmer's File Editor(Dr.REFERENCE)

プログラマ向けに特化されている多機能エディタ(図4)．NT用の実行ファイルも同梱されている．2バイト文字も扱える．

主な特徴は次のとおり．

▶仮想メモリで許される大きなファイルを編集可能．

▶複数のファイルをドラッグ&ドロップして開くこともできる．

▶よく使う機能をボタンで定義しているので，操作性がよい．

▶画面下部にステータスバーを表示し現在の設定を一目で確認することができる．

▶選択したテキストをドラッグ&ドロップ可能．

▶改行にLFを使用しているUNIXフォーマットのファイルを読み書きできる．

他にもプログラマ向けに機能強化されているので，**README.TXT**を読むほうがよい．

| ソフト名 | **Programmer's File Editor V.0.04.009**(16ビット版) |
|---|---|
| 作者 | Alan Phillips |
| 分類 | Freeware |
| 動作環境 | Windows 3.1 |
| 動作確認 | NEC PC-9821Ce+NEC製MS-Windows 3.1 |
| 所在 | **DR.REFERENCE CD-ROM ¥WIN_PROG¥PFE 00409.ZIP** |

〈図4〉Programmer's File Editor

「動作が軽快だから，メモ帳の代わりに拡張子**.TXT**や**.DOC**のファイルを開くように関連付けるだけでも非常に便利」というのが作者の弁．

● Tips/tricks for Turbo C(WIN EXPERT CD-ROM)

Turbo Cに関するテクニカルインフォメーション・ファイル．ZIP圧縮された96個のファイルから構成されている．中身はテキストデータである．

内容はソースコード，パッチ，解説書となっている．

| ソフト名 | **Tips/tricks for Turbo C** |
|---|---|
| 作者 | 不明 |
| 分類 | 不明 |
| 動作環境 | ＊.ZIPファイルを解凍し，テキストファイルをみる事のできる環境 |
| 所在 | **WIN EXPERT CD-ROM ¥PROGRAM¥T_BCINFO.ZIP** |

低価格 DOS エクステンダの登場

文/船江明彦

# Borland PowerPack for DOS16

4月中旬に発売された Turbo C++ 4.0J for DOS は，一つのパッケージで PC98 と DOS/V の両方に対応した DOS 用の開発環境でした．同時発売された Borland PowerPack for DOS16 は，DOS エクステンダ機能およびキャラクタベースの UI を構築するクラスライブラリ Turbo Vision 2.0 を同梱したお買い得パックです．

パソコンの標準 OS が MS-DOS から Windows へ移行しつつあることは，単に GUI への移行だけではなく，実行プログラムサイズの大きなアプリケーションにとって，一般的な MS-DOS の 640 K バイトのバリアが足かせになっていたことも原因の一つでした．

しかし，メモリサイズだけに着目すれば DOS エクステンダという解決方法もあります．ただし，「DOS エクステンダ＝高価」「DOS エクステンダ＝難解」というイメージがあり，使用するのに二の足を踏んでいた方も多いと思います．

## ● 低価格で実用的な DOS エクステンダが登場した！

ここで紹介する Borland PowerPack for DOS16 (以下 DOS16) は，ボーランド社の C コンパイラ用の DOS エクステンダです．

Borland C++ 4.0J または，Turbo C++ 4.0J (**写真 1**) と併用して 16 ビットの DOS エクステンダとして使用できます．マニュアルには 32 ビット DOS エクステンダの記述があるので，「あれ？」と思わせるのですが，製品の README ファイルに記述されているように，製品にはあくまで 16 ビットの DOS エクステンダだけが含まれています．ボーランド社によれば「32 ビット」の製品化については検討中とのことです．

## ● DOS エクステンダの有益性

DOS エクステンダを使用すると MS-DOS で動作するプログラムのコードサイズ，データサイズを飛躍的に増やすことができます．

そのため大きなプログラムを作ることだけに着目しがちですが，小さなプログラムでも大きな配列を使用して，メモリ上で多くのデータを取り扱う場合に有効です．

作成したプログラムはプロテクトメモリ上で動作するので，それなりに制約(決まり事)がありますが，使いこなせば非常に有益です．

## ● DOS エクステンダを使ってみる

**リスト 1** に 16 ビットの DOS エクステンダで作成した簡単なプログラム例を示します．このプログラムを実行すると，**malloc( )** などで使用できる空きメモリ数を表示します．PC9801BX2＋MELWARE でプロテクトメモリの空きが 4.2 M バイト時に約 4 M バイトの空きを表示しました．

このように DOS エクステンダを組み込んだプログラムは，ライブラリ関数も DOS エクステンダ用に置き換わり，メモリを確保する **malloc( )**，未使用領域を返す **coreleft( )** などはプロテクトメモリ用になります．

**リスト 2** では 2 M バイトの配列を確保し，0 から順に値をセットし，最後にその値を加算するものです．16 ビットの DOS エクステンダなので 64 K バイトのセグメントの制約はそのまま存在します．

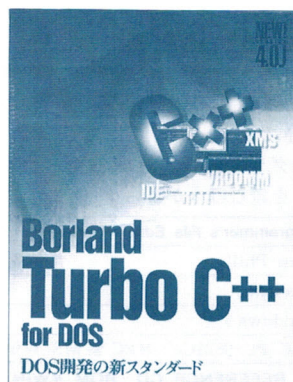
また，メモリモデルは **large** モデルに限定されるため 64 K バイトを越える配列は定義できませんが，**huge** ポインタを用いることによってアクセスすることはできます．

このように，大きな配列は動的に確保することで，後は普通の配列と同じように使用できます．

また，DOS16 はデータだけではなく，640 K バイトを越える大きなコードも生成可能なので，メモリを気にせずにいくらでも関数が定義できます．

メモリ不足に悩んでいる方は，下手な小細工をするよりも DOS エクステンダで根本的な解決をしたほうが賢明かもしれません．

## ● DOS プログラマもプロテクトモードでのコーディング知識を知っておきたい

DOS16 で作成したコードはプロテクトモードで実

〈写真 1〉 Turbo C++ 4.0J for DOS

行されるため，プログラミング時に，プロテクトモードを意識したコーディングが必要です．

例えば，**リスト3**のプログラムではセグメントレジスタESにセレクタ以外の値を代入するものですが，このプログラムを実行すると例外が発生します．

発生した例外はDOSエクステンダによってトラップされ，エラー情報を表示して終了します．

このようにプロテクトモードのプログラムでは，ES, DSなどのセグメントレジスタの取り扱いが，リアルモードに比べて異なります．

プロテクトモード時のセグメントレジスタはセレクタを示し，そのセレクタはCPUが管理するディスクリプタテーブルのインデックスのようなもので，物理メモリのアドレス，長さ，書き込み可能などの属性を保持します．

したがって，セグメントレジスタにセレクタ以外の値をセットしたり，ディスクリプタテーブルが示す長さ以上のオフセットを参照したり，読み込み専用セグメントに書き込みを行った場合，一般保護例外(GP例外，単に例外とも言う)となり，プログラムは異常終了します．

DOS16のマニュアルにはこのようなプロテクトモ

〈リスト1〉プロテクトメモリの空き容量を表示するプログラム

```
#include <stdio.h>
#include <alloc.h>
main() {
  long m = coreleft();
  printf( "¥n空きﾒﾓﾘ %ld ｷﾛﾊﾞｲﾄ" ,m/1024);
}

実行結果

空きﾒﾓﾘ 4001 ｷﾛﾊﾞｲﾄ
```

〈リスト2〉プロテクトメモリに配列を確保するプログラム

```
#include <stdio.h>
#include <alloc.h>

const long NUM = 1024L*500       ;// 500K分

main() {
   long huge *bf;          //64Kを越える配列を指すのでhugeで定義
   long m,t=0;
   bf = (long huge *)farcalloc(NUM,4);   //500K*4 2Mﾊﾞｲﾄ確保
   if (bf==NULL) return -1;              //alloc失敗
   for (m=0;m<NUM;m++) bf[m]=m;          //2Mの配列に値をｾｯﾄ
   for (m=0;m<NUM;m++) t+= bf[m];        //ｾｯﾄした値の合計求める
   printf("¥n合計 %lu",t);               //合計表示
   farfree(bf);                          //確保領域解放
}

実行結果

合計 2222725120
```

〈リスト3〉例外が発生し異常終了するプログラム

```
void main()  {
  _ES = 0x1234;      //ｾｸﾞﾒﾝﾄにｾﾚｸﾀ以外を代入
}
```

(a) ソースリスト

```
      UNHANDLED EXCEPTION 0D at 0E7F:0007 Error code: 1234

AX = 000D   BX = 0660   CX = 2AD2   DX = 0660
SI = 064E   DI = 0660   BP = 0FA6   SP = 0F98
CS = 0E7F   Limit =000C segment #02 of Z:¥DEVLOP¥CQ¥PW¥L3.EXE
DS = 0EAF   Limit =2AFE segment #05 of Z:¥DEVLOP¥CQ¥PW¥L3.EXE
ES = 0EAF   Limit =2AFE segment #05 of Z:¥DEVLOP¥CQ¥PW¥L3.EXE
SS = 0EAF   Limit =2AFE segment #05 of Z:¥DEVLOP¥CQ¥PW¥L3.EXE
- 0E7F:0000   45 55 8B EC B8 34 12 8E  C0 5D
```

(b) 異常終了した直後のメッセージ

ードのプログラムの作成方法が記述されているので一読をお勧めします．

ちなみに，Windowsでも当然プロテクトモードでプログラミングを行います．もう，86系のプログラミングでは，プロテクトモードのプログラミングが常識になってきたように思います．

なお，プロテクトモードのプログラムのデバッグには，付属のTurbo Debuggerが使用できます．Turbo Debugger上でデバッグするとエラー箇所で停止し，エラーの原因を解析できます．

**● MS-DOSのシステムコールとBIOSルーチンのコール**

DOSエクステンダということで，DOS16でもMS-DOSのシステムコール(**INT 21H**)やBIOSルーチンのコールをサポートしています．

つまりリアルモードとプロテクトモードの切り替えや，必要なデータのリアルモード-プロテクトモード間のコピーを行ってくれます．

なお，DOSエクステンダで実際に使用できる**INT 21H**やBIOSルーチンはマニュアルで調べる必要があります．なお，PC98のBIOSルーチンのコールはサポートされていないようです．

同じように，C言語で標準に提供される関数の一部も使用できないので，やはりマニュアルで調べる必要があります．

**● DPMIのサポート**

前述のように，**INT 21H**やBIOSルーチンのコールはDOSエクステンダでエミュレーションしてくれますが，独自のドライバ，割り込み処理をDOSエクステンダのプログラムから使用する場合は，DPMI(DOS Protect Mode Interface)サービスを使用します．

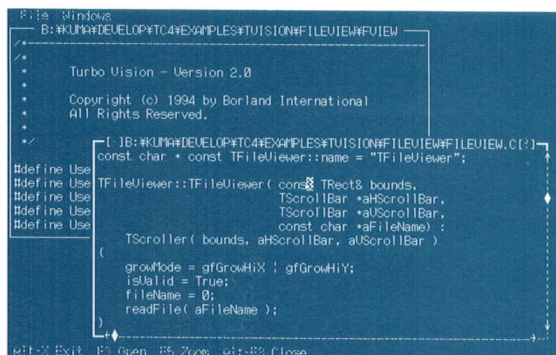
DPMIサービスはWindowsでもサポートされ，プロテクトメモリの管理，プロテクトモードとリアルモードの切り替えなどの重要な機能をサポートします．マニュアルにはDPMIファンクションについて詳しく説明が記述されています．

**● Windows API，DLLのサポートでDOS/Windows共用DLLが作成できる**

驚いたことにDOS16はWindowsの一部のAPIをエミュレートします．また，Windowsで使用しているDLLの呼び出しとDLL自体の作成ができます．

当然GUI部分などはサポートされていませんが，まとまった処理などをDLLで作成すれば，DOSとWindowsの両方で使用できるモジュールが作成できそうです．

また，エミュレータがサポートしていないAPIをフックして，自分でエミュレーションする方法などがマ

〈図1〉Turbo Vision 2.0で作成したテキストビューワ
(表示しているのはソースコード)

ニュアルに記述されているので，応用範囲もかなり広いと思われます．

またREADMEファイルにはWindows用のDLLで提供される同社のParadoxエンジンの呼び出しについての記述があり，ボーランド社内部の開発ツールを外販しているような感じがみられます．

**● プロテクトモードで動作するTurbo Visionが付属**

付属するTurbo Vision 2.0は，イベント駆動型のアプリケーション作成ツールです．ちょうどWindowsをテキストベースにしたようなものがMS-DOSで動作すると考えてよいでしょう(**図1**)．ユーザーインターフェースは，Turbo C++の統合環境などと同じものです．

DOS16にはプロテクトモードで動作するTurbo Visionがソースコード付きで付属します．またグラフィックライブラリであるBGIもプロテクトモード用がサポートされています．

**● その他の注意点**

16ビットのDOSエクステンダで作成したプログラムは実行時に付属の**RTM.EXE DPMI16BI.OVL**が必要です．またBGIを使用する場合はBGIドライバ，フォントと**BGI16.DLL**が必要です．

*　　　*　　　*

筆者は仕事を探しています．
よろずソフト開発承ります．
連絡先☎078-753-6443 船江まで．

製品名／価格　Borland PowerPack for DOS16／18,000円
　　　　　　　Turbo C++ 4.0J for DOS／29,800円
問い合わせ先　ボーランド㈱
☎03-5350-9380

# インターネットアクセス・レポート⑦

《www.atitech.ca, www.diamondmm.com》

## ● グラフィックカードのサポート

秋葉原のDOS/Vショップをまわると，PCIバスのグラフィックカードが人気で，VLバスを採用しているカードは安く売られているところが多く見られます．まして，ISAバス用のカードはほとんど店頭にありません．

しかし，世界中で使用されているAT互換機にはISAバスしかないマシンが非常に多いので，米国などのグラフィックカードを作っている会社では，最新のアクセラレータチップを使っていても，ISAバス,VLバス,PCIバス用の製品ラインナップをそろえていることが多いようです．

## ● グラフィックボード・メーカーのWWWのサイト

この業界の中でも古くからオリジナルのアクセラレータチップを使ってグラフィックボードを作っているカナダの会社にATI TECHNOLOGIES社がありますが，WWWおよびFTPサイトを運用しています（**www.atitech.ca**, **ftp.atitech.ca**）．

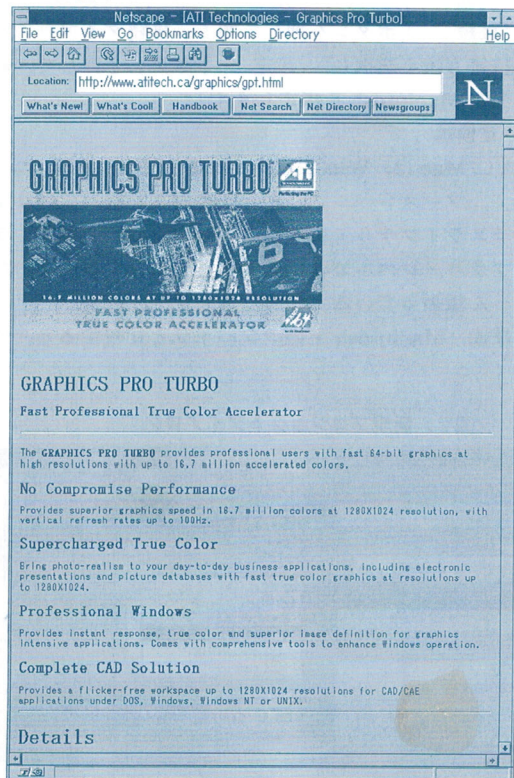
最近では，PCIバスを採用したApple社の最上位モデル9500シリーズのビデオボードに同社のアクセラレータカードが使われています．

また，日本でも人気の高いグラフィックカードのメーカーであるDiamond Multimedia Systems社も**www.diamondmm.com**というWWWサーバーを動かしています．

その他，日本ではあまり製品を見かけませんが，STB社も**www.stb.com**を上げているようですが，つながったことがありません．

日本国内でよく出回っているNumber Nine Visual Technology, Matrox Graphics社はWWWサーバーをまだ上げていないようです．

また，国内ではメルコ社がWWWサーバーを上げています（**www.melcoinc.co.jp**）．最近は年間約50人の新規社員を採用している同社ですが，WWWサーバー内にも求人案内があり，履歴書を入力する画面も用意しています．

▼ATI社のWWW

▼Diamond社のWWW

Windows上でマックのアプリケーションを開発できるツール

文/濱田功志

# Microsoft Visual C++ Cross Development Edition for Macintosh

Visual C++ cross development edition for the Macintoshは，Macintoshのアプリケーション開発のためのオプション開発キットです．日本語には対応せず，PowerPCコードの出力もできないため，本格的な開発ツールとして用いることはできませんが，Windows(Win32)用とMacintosh用のアプリケーションを単独のプロジェクトから生成できる，魅力的な開発環境です．

## ● VC/Macの開発環境

VC/Macを使ったプログラムの開発は，基本的にWindows NT上で動作するVisual C++ 2.0上で行います．そのため，VC/Macのパッケージの中には，Visual C++ 2.0は含まれていませんので，別途入手する必要があります．

筆者は，Visual C++の日本語版に組み込んで使用してみましたが，特に問題なく使えました．ただし，正式なサポートを受けるためには，英語版のVisual C++ 2.0と組み合わせることが前提になるでしょう．

VC/Macで作成したアプリケーションを実行/デバッグするためには，当然ながらMacintoshが必要になります．そこで，コーディングを行うWindows NTマシンとMacintoshは，イーサネットもしくはシリアルケーブルで接続します．そして，Macintosh側にはファイル転送ユーティリティと，デバッグユーティリティをインストールします(図1)．

イーサネットで接続する場合は，Windows NT側でAppleTalkプロトコルを動作させます．

VC/Macのインストールは，あらかじめVisual C++ 2.0がインストールされているWindows NTマシンに対して行います．インストールが無事に終了すると，プロジェクトのターゲットとして，「Macintosh」が選択できるようになります(図2)．

また，Macintosh側には，フロッピディスクから上記の2種類のユーティリティをインストールします．

Windows NT上でビルドされたファイルは，イーサネットまたはシリアルケーブルからMacintoshに転送されます．また，Windows NT上で，Macintoshのアプリケーションを制御したり，レジスタを参照したりしながらのデバッグを行うこともできます．そのための設定は，双方で行う必要があります．

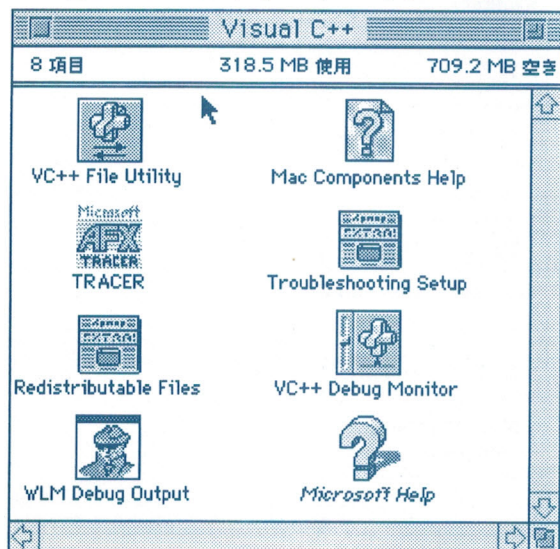
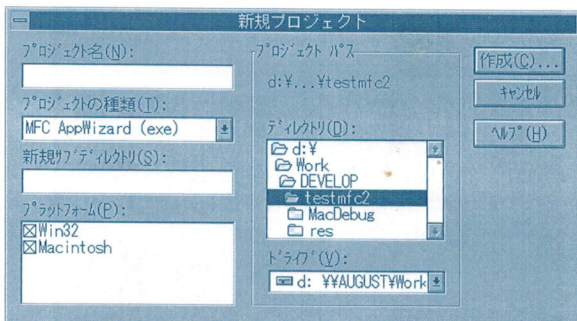
Windows NT側は，[ツール]-[オプション]メニュー内のデバッグオプション・ダイアログによって行います(図3)．

[接続...]ボタンを押して開く，接続ダイアログから接続方法を選択し，[設定...]ボタンを押して，転送先のMacintoshを指定します．

## ● 互換性

VC/Macは，Win32互換ライブラリと，MFCによって，ソースレベルでの高い互換性を実現しています．クラスライブラリ・レベルでの互換性だけでなく，ライブラリ・レベルでの互換性を実現している点は，たいへん頑張っていると評価できます．

反面，Macintosh上のコンパイラと比較すると，生

〈図1〉Macintoshにインストールするツール群

〈図2〉新規プロジェクトダイアログ

成したコードのパフォーマンスは，決して有利ではないでしょう．

APIについては，リファレンスマニュアルにサポート/非サポートが明記されていますので，マニュアルを参照しながらの開発になるでしょう．

サポートされていないAPIに関しては，**#ifdef**などを使い，ターゲットによるコードの差し替えを行う必要があります．例えば，DIBの表示などはサポートされていますが，**sndPlaySound( )**などの，音関係のAPIはサポートされていません．

また，Cで記述したプログラムを移植する場合は，Aboutダイアログのメニュー位置の違いなどを，ソース上で修正します．

AppWizardを使用した場合は，そのような外観上の違いも自動的に反映されますので，特に意識する必要はありません．

MFCおよびAppWizardを使用した場合は，OLEやODBCなどのクラスライブラリがサポートされません．ただし，その他の部分では非常に高い互換性を持っており，生成されたスケルトンは，無変更でMacintosh上で実行させることが可能です．

ただし，メニューやダイアログボックス上の日本語は，全て文字化けしてしまいます．単にフォントの問題だけでなく，文字コードそのものが化けていますので，単純に解決はできません．

ダイアログボックスやメニューなども，Macintosh標準のリソースには格納されておらず，独自の方法でメニューが構築されているようなので，ResEditレベルで修正することもできませんでした．

どうしても，日本語を表示したいのであれば，実行時にメニューアイテムを置き換えていく必要があるでしょう．

メニューやダイアログのリソースは，実現の方法が異なるようですが，ストリングテーブルだけは，**STR#**リソースとして格納されています(ただし，やはり日本語は文字化けします)．したがって，次のような手順で，日本語を取り扱うことができます．

▶使用したい日本語文字列をストリングリソースに格納する．その際に，判別できるように番号を振るか，英語表記にする．

▶実行プログラムを作成する．

▶Macintosh上で，ResEditを用いて，完成したプログラムを開く．

▶**STR#**リソース内の文字列を，日本語に置き換える．

この方法では，プログラムをビルドするたびに，ResEditで文字列を入力しなければなりませんので，現実的な方法とはいえません．

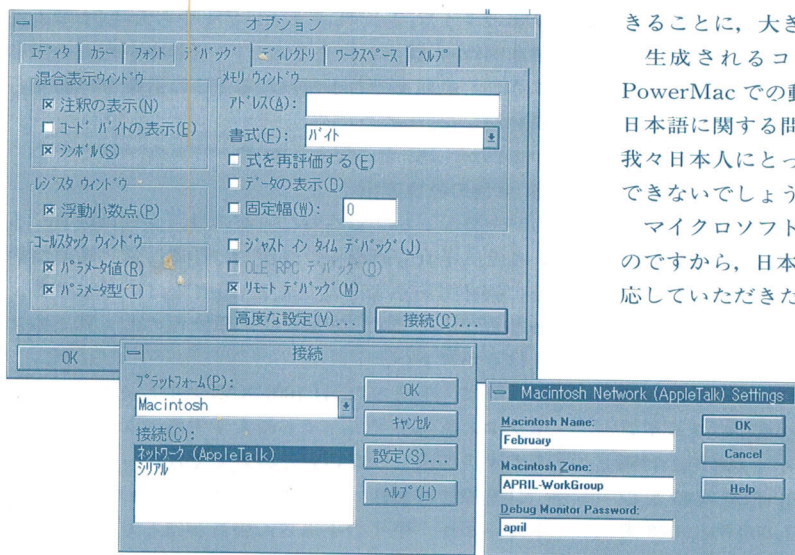
VC/Macで，日本語アプリケーションを開発するのは，ちょっと手を出さないほうがよさそうです．

### ● まとめ

マルチプラットフォームに対応したアプリケーションは，ユーザーにとってもメーカーにとっても魅力的で有益なものです．これまでは，DirectorやAuthorwareなどのオーサリングツールやデータベースアプリケーションでしか実現されなかったものが，C/C++言語製品でソースレベル互換な開発環境が提供されたことは，たいへん有意義なことだと思います．

開発者にとっても，単独の環境でMacおよびWin32の開発を管理できる点は，かなり魅力的でしょう．筆者自身も豊富な機能を持つ(構文の色分け，ドラッグ&ドロップ，オンラインヘルプの充実など)VC++ 2.0のエディタに使い慣れていますので，同じ環境下でMacintoshのアプリケーション開発ができることに，大きな魅力を感じています．

生成されるコードは，68Kコードであっても，PowerMacでの動作に問題はなさそうです．ただし，日本語に関する問題点はどうしようもなく，現状では，我々日本人にとっては一部の限られた開発にしか利用できないでしょう．

マイクロソフト社は，ExcelやWordで実績があるのですから，日本語/OLE2/PowerPCなどに，ぜひ対応していただきたいものです．

〈図3〉◀
接続オプションの設定

# テクノロジエッセンス・レポート⑥

《アジアの時代 台湾の時代(後編)》

前回は,台湾(R.O.C)のコンピュータ関連企業の印象を紹介した.たまたま訪問した時期に,台湾のコンピュータショー Computex が行われていたが,本誌が発売されるころには,すでに他誌で取り上げていると思う.

そこで,ここでは,台湾(台北市)のコンピュータショップ街「光華商場」について簡単にリポートしたい.

## 台北市の電脳街 光華商場

光華商場は,東京の秋葉原電気街のコンピュータショップとパーツショップを集めたようなところだ.

台北市にたどり着いてから,光華商場に向かうのは,タクシーを使うのが一般的だ.中国語が達者ならばバスでもよいが,タクシーの初乗り運賃が50元(NT$50≒180円)なので,たいした金額ではない.場所は**図1**に示した.

タクシーに乗ったら(ドアは手動),運転手に光華商場(guangh-wa-shang-chang)と言えばよい.

## PCと清潔なパーツショップ

パソコンはどこのショップもIBM PC.なお,Mac(Apple)のショップは,1軒だけしか見つけられなかった.

主な製品の価格(5月末)は,P5-75マシン:143,000円(PCI, I/O付き,RAM:8Mバイト,HDD:540Mバイト,ET-4000(PCI),14"モニター),DX4-100マシン:102,150円(VL,RAM:4Mバイト,HDD:540Mバイト,VL-5428,14"モニター),P5-75マザーボード:71,000円(CPU,RAM:8Mバイト付き),DX4-100マザーボード:39,900円(CPU,RAM:4Mバイト付き),倍速CD-ROMドライブ:14,000円,486SLC-50ノートパソコン:100,000円(HDD:130Mバイト RAM:4バイト),486DX2-66ノートパソコン:176,000円(HDD:250Mバイト,RAM:4Mバイト),486DX2-66ODP:13,600円,DX4-100ODP:21,500円,DX4-100:20,000円,AMD DX2-80:11,500円,AMD DX4-100:16,100円,P5-100/120:64,500/89,600円,NECマルチシンクモニター2v(14インチ):34,000円,MS Excel/MS Word(中文版):各39,000円,Seagate HDD 720M/1.1Gバイト:22,000/32,000円,CD-ROM(画像データ集):1,400円,DSub 9ピン(カバー付き):180円,マウス:540円,3.5インチ 2HD(10枚):650円,マウスパッド:100円,8255実験ボード(ソフト付き):3,400円,PLD実験基板(PAL, GAL, PEEL, EPLD, PLA):7,000円,8048実験基板(ソフト付き):4,300円,8051実験基板(ソフト付き):9,300円,はんだごて:320円から1,900円のものまで,自販機の緑茶(250 ml):35円〔以上1元(NT$1)=3.58円で計算〕.

〈図1〉
光華商場の場所
(網かけの部分)

光華商場周辺には,パソコンの周辺小物機器を含んだパーツショップがある.第1印象としては,秋葉原のパーツショップに比べて,清潔感が漂っていること.

ただし,ジャンク屋もあって,ゴミ同然に扱われているMADE IN USAの286マシンがあった.一応売り物のようで,聞いてみると,どうやら大陸(中華人民共和国)から買出しに来るらしい.世界中をたらい回しにされ,最期までこき使われるPCの宿命を垣間見る思いがした.

これが,PCなんだよ,98君.

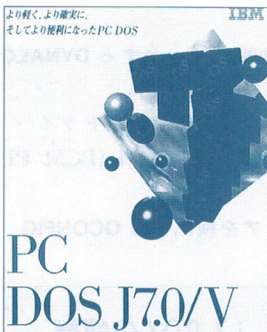
## 女性の社会進出

パーツショップで意外だったのは,女性の店員が多いこと.それも単なるレジ係ではなく,客の質問に的確に答えているように見える(中国語だから正確にはわからない).また,客層に占める女性の数も秋葉原よりはるかに多い.

試しに,手持ちのHP100LXのシリアルポート用コネクタの有無をアルバイト&大学生風の女性に聞いてみたが,即座に「ない」と答えてくれた.

秋葉原で「女性」と言えば,ビラ配りのミニスカートギャルか,中年の小母さん店員がほとんどだと思うから,かなり対照的だった.

DOSのバージョンアップはつづく

文/山下直彦

# IBM PC-DOS J 7.0/V(日本語版)の特徴

PC-DOSの新バージョンが発表されました．本稿が掲載される頃には発売(8月9日予定)されているでしょう．試用版を入手しましたので，その概要について紹介しましょう．なお，製品版との相違があるかもしれませんのでご留意ください．また，英語版PC-DOS Ver 7.0についての紹介〔参考文献(1)〕もあわせて参照してください．

● **PC-DOS J 7.0/Vに関する製品情報**

PC-DOS J 7.0/Vは，新規購入(消費税別14,000円)の場合は3.5インチのディスケットのみで供給されます．また，アップグレード版(同9,000円)には，3.5インチのディスケットとCD-ROMの2種類があります．

アップグレードの対象は，

▶ **IBM製品**：日本語DOS K 3.4，J 4.0，J 4.0/V，J 5.0，J 5.0/V，PC-DOS J 6.1/V，J 6.3/V

▶ **他社製品**：MS-DOS 5.0/V，6.2/V，Compaq DOS 5.0/V，DR DOS 6.0/V

などです．そして，環境としては次のものが最低必要です．

▶ハード・ディスク：10 Mバイト～27 Mバイト

▶ディスケットドライブ　：1.44 Mバイト3.5インチ(インストールに使用するときに必要．特殊なドライバなしに使用できること)

## 英語版と共通な特徴

英語版のPC-DOS Ver 7をベースにしているので，英語版の特徴は，ほぼそのまま引き継がれています．主要なものについて説明していきます．

**(1) セットアップ・ディスクの拡張フォーマットXDF**

ディスケット版の場合，英語版は5枚，DOS/V版は9枚で供給されます．いずれも一枚目は2HD 1.44 Mバイトの通常のフォーマットになっていますが，2枚目以降はXDFフォーマット(注1)が使用されています．

DOS/Vの試用版では，2枚目および9枚目のディスケットには，バンドルファイルに含まれない単独ファイルもいくつかありました．

セットアップ・ディスクの内容は，解凍後ハードディスクにインストールされます．なお，PKWARE社のPKLITE〔参考文献(2)〕による実行時解凍ファイルも多数見いだせます．

**(2) 基本メモリの使用量の削減**

**HIMEM.SYS, EMM386.EXE, ANSI.SYS, DOSKEY.COM**や**MOUSE.COM**などのTSRの基本メモリへの常駐量が削減されており，これは**DOSDATA=**コマンドや，**COMMAND.COM**の**/h**スイッチの導入，メモリ最適化ツール**RAMBoost**の改良などによって実現されています．

DOS/V版では，必要なデバイスドライバが多いことから，UMBが比較的簡単に使い切られる事情もあるので，今後の使用例でその有効性を確認していきたいと思います．

**(3) Stacker 4.0**

SuperStor/DSに代わるディスク圧縮ツールとして，Stacker 4.0が導入されました．これは，ほかの圧縮ツールに比較し，圧縮効率がすぐれていることと，そのドライバ**STACHIGH.SYS**にDPMS(DOS Protected Memory Services)を利用し，基本メモリの消費量を削減しています．

**(4) Eエディタの改善**

マウスの使用，プルダウン・メニューのサポート，読み取り専用での利用(**Browse.com**)などの改善がなされました．また，REXXやCのキーワードの一部が入力されると残りを自動生成する機構が導入されました．

これは，OS/2のエディタであるEPMの場合と同様の機能です．REXXの場合は，拡張子が**.bat**あるいは**.cmd**で行頭がコメントの/*で始まるとき，Cの場合は拡張子が**.C**，**.CPP**，**.H**あるいは**.HPP**のとき，この機能が動作します．

例えば，**if**を入力した後にスペースを押すと，前者では，

```
if then
else
```

が生成され，また後者では，

```
if ()
{
}
```

(注1) **XDF.COM**を常駐させてはじめて内容を見ることができる．この時，23セクタ×80トラック×2ヘッド×512バイト=1840 Kバイトで扱われる．また，XDFフォーマットのディスケットのバックアップコピーには**xdfcopy.exe**を使用する．

〈図1〉 find でのワイルドカード文字の使用例

```
C:\DOS>find "[" D*.ini

---------- DOSSHELL.INI
[savestate]
[programstarter]

---------- DOSV.INI
[SYSTEM]
[FONT]
[DISPLAY]
[KEYBOARD]
[INPUT]
[PRINTER]
```

**else**

{

}/＊ **endif** ＊/

が生成されます．

内部コマンドで **EXPAND OFF** とするとこの機能は抑止され，Ctrl+**X** を押すと動作します．ほかにも多くのカスタマイズ項目があり，**E.INI** ファイルで設定できます．

また，コマンドライン・スイッチやその一部は環境変数 **E** で設定できます．

**(5) コマンド Find の拡張**

ファイル名の指定に，ワイルドカード文字＊や？が使えるようになり，複数ファイルにわたる検索ができるようなりました．使用例を**図1**に示します．さらに，スイッチ **/S** が追加され，サブディレクトリを含めた検索も可能となりました．

**(6) その他の特徴**

種々のコマンドツールが拡張・追加されました．

▶ファイル更新ユーティリティ **FILEUP**

▶モービルコンピュータのドッキングをサポートする **DOSDOCK**

▶デバイスドライバを動的にロードする **DYNALOAD** コマンド

▶より効率的な省電力を実現する APM ドライバ

▶Phoenix PCMCIA サポートソフト(PCM PLUS V 3.01)

▶より多くのハードウェアを検出する **QCONFIG**

などです．

## PC DOS J7.0/V の特徴

### ● DBCS 対応の REXX のサポート

REXX はインストール時の選択項目(デフォルトではインストールされない)になっています．英語版の PC-DOS Ver 7.0 からサポートされましたが，DOS/V 版の日本語モードでは，2バイト文字セット(以下 DBCS と表す)対応となりました．具体的には，キーワード命令 **Options** と **DBCS** 関数の追加です．

**Options** の構文は下記のとおりです．

**OPTIONS** ***expression*** **;**

ここに ***expression*** として，次の用語が入ります．

▶ **ETMODE**…DBCS の文字を含む文字列をプログラム中で使用すると指定．

▶ **NOETMODE**…上記を使用しないと指定．

▶ **EXMODE**…(DBCS と1バイト文字セットの)混合文字列中の DBCS データは，命令，演算，関数によって論理文字単位(DBCS 2 バイトを1文字)で処理されるように指定．

▶ **NOEXMODE**…上記を1バイト単位で処理されるように指定．

英語版の PC-DOS Ver 7.0 の REXX では，マニュアルには **Options** の記述はありませんが，実は REXX

〈リスト1〉 DBCS 関数の使用例

```
/* DBCS.BAT        By N.Yamashita */
Options 'ETMODE'
mode.1='EXMODE'
mode.2='NOEXMODE'
do  i=1 to 2
   Options mode.i
   Say '['i'] Ｎｏｗ  ｉｎ  'substr('１２ＮＯＥＸＭＯＤＥ',5)
   Say substr('ＡＢabＣＤ',4)
end /* do */
exit
.


/* DBCS.BAT        By N.Yamashita */
Options 'ETMODE'
mode.1='EXMODE'
mode.2='NOEXMODE'
do  i=1 to 2
   Options mode.i
   Say '['i'] Ｎｏｗ  ｉｎ  'substr('１２ＮＯＥＸＭＯＤＥ',5)
   Say substr('ＡＢabＣＤ',4)
end /* do */
exit
```

〈図2〉 リスト1のプログラムの実行結果

```
[1] Ｎｏｗ  ｉｎ  ＥＸＭＯＤＥ
bＣＤ
[2] Ｎｏｗ  ｉｎ  ＮＯＥＸＭＯＤＥ
aabＣＤ
```

解釈プログラム **REXX.EXE** 内の内部データ領域には，**Options, ETMODE, NOETMODE, EXMODE, NOEXMODE** などの文字が見られます(どのようにして調べ得るかについてはここでは述べない．別の機会に譲る)．すなわち，DBCS 対応を見越して開発されたことがうかがえます．

DBCS 関数としては，**DBADJUST, DBBRACKET, DBCENTER, DBLEFT, DBRIGHT, DBRLEFT, DBRRIGHT, DBTODBCS, DBTOSBCS, DBUNBRACKET, DBVALIDATE, DBWIDTH** があります．

例として，**リスト 1** に示すプログラムを実行して見ましょう(**図 2** 参照)．

2 行目の **options 'ETMODE'** は書かなくても動きますが，安全のため，必ず含めるという習慣を付けておいたほうがよいでしょう．**do** ループで同じコマンドを一度目は **EXMODE** で二度目は **NOEXMODE** で実行してみるという例題です．

第 3 パラメータ以降を省略した **SUBSTR** は第 1 パラメータの文字列から第 2 パラメータで指定された番目の文字以降を返します．

**EXMODE** では全角文字，半角文字の区別をしないので，

- ▶ 7 行目の用法では **'EXMODE'**
- ▶ 8 行目の用法では 4 番目の文字は **'b'** となり **'bCD'** が返されます．

一方，**NOEXMODE** では，1 バイトを 1 文字と勘定しますので

- ▶ 7 行目の用法では **'NOEXMODE'**
- ▶ 8 行目の用法では 4 番目は **'B'** の第 2 バイト(シフト JIS 表現で **'B'** は **82h 61h** 故) **61h** から始まります．したがって，**'aabCD'** が戻されます．

なお，同じプログラムを英語モードで実行すると当然，文字化けを起こしますが，両モードで同じ出力となりますので，**Options** は英語モードで対応していないことが確認されます．

## ● View コマンド

**View** コマンドでオンラインマニュアルを読むことができます．日本語モードでは，**README.INF, CMDREF.INF, DOSREXX.INF, DOSERROR.INF** が該当します．また，英語モードでは **README.INF** を除くものが扱えます．**.INF** に対するパス指定を行えば，日本語モードから英語モードのオンラインマニュアルにアクセスすることができます．通常の設定では，例えば，

**View　c:¥dos¥ps2¥cmdref**

と入力することにより，英語モードのコマンド解説にアクセスできます(あるいは，**CHEV US** で，一時的に英語環境に切り換えた後，**View cmdref** を入力してもよい)．

日本語モードと英語モードのオンラインマニュアルは単なる翻訳でない面もあり，相補的要素があります．したがって，一度は目を通されることを推奨します．例えば，英語モードの **CMDREF.INF** の中にある **What's New for PC DOS 7** には，従来の DOS との相違点についての要約があります．

PC-DOS の Viewer は若干の制約があるものの，OS/2 の Viewer と互換性があります．したがって，PC-DOS Viewer で OS/2 の **.INF** ファイルを表示したり，逆に OS/2 の Viewer で PC-DOS の **.INF** ファイルの表示を行うこともできます．

## ● マニュアルに記載されていないユーティリティ

以前から，PC-DOS にはマニュアルで触れられていないコマンドなどが多数ありました．日本語モードの **README.INF** あるいは同じ内容の **README.TXT** の最後のほうに，上記のタイトルのもとに **EDLIN, COMP, KKCFUNC, ESCP12, HELP5, INSTANCE, FDISKUS, 32705250, IPLDRIVE, HARDBOOT** などのコマンドの簡単な説明が述べられています．

これらは以前のバージョンからあるコマンドですが，例えば，**INSTANCE.COM** は **RAMBoost** によって UMB メモリにロードされたデバイスドライバを Windows のローカルなデバイスドライバ(インスタンス)として設定するときに使用するものです．このコマンドなどはマニュアルに入れてもよいと思います．

なお，以前のバージョンからありますが，**KEYBIBM.COM, KEYBAX, COM, KEYBJ3.COM** が各々のキーボード用として含まれます．ただし，これらのヘルプファイルは見あたりませんでした．

◆ 参考文献 ◆


(1) 山下直彦；IBM PC-DOS Ver 7.0(英語版)の特徴，トラ技コンピュータ 7 月号，1995 年．

(2) 山下直彦；デバッガの効果的活用法(第 7 回)，トラ技コンピュータ 2 月号，1994 年．

## ● 最高速 Pentium CPU のパーソナルコンピュータ

インテルの最速 Pentium CPU(133 MHz 版)を搭載した各社のハイエンドモデルが出揃った.

NEC は，新 98MATE X で CPU とグラフィック処理能力を高速化して動画再生性能を大幅に向上させている.

同時に発売した「デジタル動画エンコーディングセット」(圧縮ボード，エンコーダコントローラソフト RealMagic Producer，デジタルビデオ編集ソフト Adobe Premier を含んで 650,000 円)とともに，マルチメディアタイトルの制作/利用の環境を整えた.

それに対して新 98FELLOW は今現在必要のない機能を削ぎ落として低価格を維持した.

なお，Windows 95 への移行が順調に進むのか懸念されていた PC98 シリーズだが，今回の新製品では「95 対応」だけでなく，推奨機能への対応も果たしている.

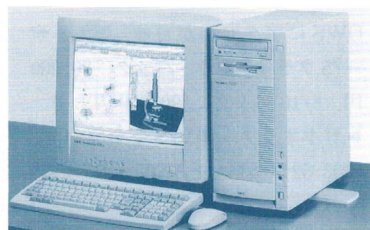
DOS/V 機は価格競争が激しく Endeavor AT/ET6500(Pentium 133 MHz)は早くも約 20 % の値下げを行い，HDD なしモデルで 235,000/245,000 円.

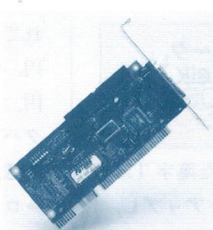
| 名称 | OptiPlex X5133 | 98MATE X シリーズ | T シリーズ | 98FELLOW |
|---|---|---|---|---|
| 型名 | OptiPlex XMT5133 | PC-9821Xt13 | T5133DT/TW | PC-9801BX4/U2 |
| CPU-MHz | Pentium-133 | ← | ← | 486DX2-66 |
| メモリ(最大) | 16 MB (8 MB〜128 MBから選択) | 16 MB(256 MB) | EDO DRAM 16 MB | 2 MB(64 MB) |
| 拡張バススロット | ISA×4, PCI/ISA×2 | PCI×2, C バス×5 | PCI×3, ISA×2, PCI/ISA×1 | C バス×3 |
| HDD | 540 MB(365 MB, 540 MB, 1 GB から選択) | 1.2 GB | 1.2 GB | オプション |
| グラフィック(VRAM) | Vision864(2 MB) | MGA-2064W(4 MB) | Mach 64(2 MB) | GD5430(1 MB) |
| 標準装備品 | 特になし | 4 倍速 CD-ROM | ← | 特になし |
| 特徴 | 最新の Pentium を搭載したビジネスモデル．最小構成(XL5133, 8 MB RAM, 365 MB HDD, 1 FDD)で 328,000 円 | 最新の Pentium を搭載し，拡張性にも優れたハイエンドのミニタワー型．Windows 95 必須機能のほか推奨機能も標準搭載. | 通信販売専用ハイエンドパーソナルユース向け．PnP 対応．DT：デスクトップ型，TW：タワー型 | CPU 速度の向上，ウィンドウズアクセラレータの標準搭載．Windows 95 の必須機能にも対応．FDD は 2 基搭載. |
| 導入済み OS, ソフト | DOS 6.2/V, Win 3.1 | DOS 6.2，Win 3.1，Video for Windows 1.1e(ランタイム版) | DOS 6.2/V, Win 3.1 | |
| 価格(円) | 390,000 | 690,000 | 368,000/378,000 | 98,000 |
| 問い合わせ先 | デルコンピュータ㈱ ☎ 0120-1984-86 | NEC パソコンインフォメーションセンター ☎ 03-3452-8000, パソコン情報サービス FAX03-3522-9801 BOX 番号 7048 | ㈱ティアックシステムクリエイト テクニカルサポート ☎ 0423-80-7266 | NEC パソコンインフォメーションセンター ☎ 03-3452-8000, パソコン情報サービス FAX03-3522-9801 BOX 番号 7013 |

| 名称 | Aptiva Vision | Aptiva 515/530/730 |
|---|---|---|
| 型名 | 2407-NF2/2407RWD | 2114-27J/70J/63J |
| CPU-MHz | 486DX2-66/DX4-100 | 486DX2-66/DX4-100/DX4-100 |
| メモリ(最大) | 8 MB(128 MB) | 8 MB(128 MB)/8 MB(96 MB)/8 MB(128 MB) |
| 拡張バススロット(使用可能数) | ISA×4(2) | ISA×4(2)/4(2)/8(6) |
| HDD | 540 MB | 540 MB/720 MB/540 MB |
| ディスプレイ | 14″CRT(一体型) | 15″CRT(外付け型) |
| グラフィック(VRAM) | GD5430(1 MB) | |
| 標準装備品 | 倍速CD-ROM, SoundBlaster16, 内蔵ステレオスピーカ，ボイス FAX モデム | 4 倍速 CD-ROM(515 のみ倍速), SoundBlaster16, 外付けステレオスピーカ，ボイス FAX モデム |
| 特徴 | 全モデルに OS/2 Warp V 3-日本語版を初期導入し，Internet への接続が容易 | 個人消費者向け Aptiva リースを開始 |
| 導入済み OS, ソフト | PC DOS J6.3/V, Win 3.1, OS/2 Warp V 3 日本語版，MS Works, FAX/音声統合ソフト，電子辞書，ほか百科事典や音楽用のマルチメディアソフト，パソコン通信や Internet 接続などの通信ソフトなど | |
| 備考 | TV/MPEG 機能搭載モデルもある(2407-RTD) | アップグレードサービス 1995 年末日まで受け付け．IBM PC Direct アップグレード係 ☎ 0120-1-55307, FAX 0120-1-55306 |
| 価格 | オープンプライス | |
| 問い合わせ先 | ダイヤル IBM ☎ 0120-04-1992，Aptiva リース：IBM PC Direct Aptiva ダイヤル ☎ 0120-80-4545，FAX 0120-87-4545 | |

## ● Warp 搭載のマルチメディア・パソコン

新しい Aptiva は，全モデルに OS/2 Warp が初期導入されており Internet へ簡単に接続できるようになった.

また今回発表のモデルと Aptiva Music Vision を対象として個人消費者向け Aptiva リースを開始した．リース期間は 24 ヶ月．そのほか PC に接続してペン入力を可能にするオプション機器「IBM Smart Tablet」(24,800 円)を発表した．小型のタブレットと電磁誘導型のペン，Windows 用のユーティリティソフトからなり，ペンによる手書きの文字や絵を画面に入力できるもの.

▲ PC9821 Xt13

▲ Aptiva 515

▲ SC-DV/ISA

▲ PCSC-II

## ● SVGA 対応のノートパソコン

ノートパソコンの表示素子は，10.4 インチサイズが定着し，SVGA 対応も増えてきた．

買ってすぐに使える多彩なアプリケーションソフトを満載し，CD-ROM にボイス FAX モデムまでも装備した高機能のマルチメディア・タイプが多く，少し前のデスクトップより高速でメモリも多く充実している．

PC9821Na7 は IrDA 対応の赤外線インターフェースをもつ．

| 名称 | CONTURA 420/430 | LIBRAGE | FMV-BIBLO サブノート | 98NOTE |
|---|---|---|---|---|
| 型名 | CONTURA 420C/430C/430CX | FN-P75DC | FMV-475NL/T，NL/S | PC-9821Na7/Nx |
| CPU-MHz | DX4-75/DX4-100/DX4-100 | Pentium-75 | DX4-75 | Pentium-75/DX4-100 |
| メモリ(最大)バイト | 8 MB(32 MB) | 8 MB(40 MB) | 8 MB(24 MB) | 8 MB(40 MB) |
| HDD バイト | 420 MB/720 MB/720 MB | 820 MB | 340 MB | 720 MB |
| ディスプレイ | 10.4″ DSTN/DSTN/TFT | 10.4″ DSTN(ビデオキャッシュメモリ 512 KB) | 8.4″ TFT，8.5″ DSTN | 10.4″ TFT/10.1″ TFT |
| グラフィック(ビデオ RAM) | ローカルバス(1 MB)．VGA 256 色 | ローカルバス(1 MB) 800×600 256 色 | GD6225，GD6235(512 KB) VGA 256 色 | Cyber9320(1 MB) 800×600 256 色/640×480 64 K 色，他 |
| サウンド機能 | | SoundBlaster PRO 互換，内蔵モノラルスピーカ | ← | PCM 録音・再生，FM 音源，ステレオスピーカ |
| サイズ(mm) | 228.3×302.9×54.9 | 296×216×45 | 260×210×48 | 316×254×59 |
| 重量 | 2.84 kg | 2.7 kg | 2.1 kg，2.0 kg | 約 4.0 kg/3.9 kg |
| バッテリ(駆動時間) | デュラセル NiMH (約 3.0～5.0 時間) | NiMH | Li イオン(約 1.5～4 時間) | NiMH(約 1.1～1.5/1.6～2.0 時間) |
| 特徴 | 光学式トラックボール，オンライン・チュートリアルの装備などイーズ・オブ・ユース機能を充実させた． | アイ・シー・エム社初のノートパソコン | ワープロと表計算ソフトを導入したモデル(418,000，328,000 円)も用意． | CD-ROMドライブとTFTカラー液晶を搭載したマルチメディア・カラーノートパソコン． |
| 導入済みソフト | DOS 6.2/V，Windows 3.1，Lotus Organizer，オンライン・チュートリアル | DOS 6.2/V，Windows 3.1 | ← | DOS 6.2，Windows 3.1，TranXit 1.0 |
| 価格(円) | 328,000/398,000/488,000 | オープンプライス | 388,000，298,000 | 660,000/560,000 |
| 問い合わせ先 | コンパック・カスタマーセンター ☎ 0120-101589 | ㈱アイ・シー・エム ☎ 06-350-5701，FAX 06-350-5711 | 富士通㈱ FM インフォメーションサービス ☎ 0120-222-015 ☎03-3646-0816，価格/カタログ請求窓口 FAX 043-299-3642 | NEC パソコンインフォメーションセンター ☎ 03-3452-8000，パソコン情報サービス FAX03-3522-9801 BOX 番号 7014 |

## ● SCSI-2 カード

| 型名 | HS-MS4/MS4T | SC-DV/ISA | IFC-NS/ES/VS | PCSC-II |
|---|---|---|---|---|
| 対応機種 | PowerMacintosh9500 | DOS/V 機 | PC98，EPSON，DOS/V のノートパソコン | ← |
| 対応バス | PCI | ISA | PCMCIA TYPE 2 | PCMCIA TYPE 1 |
| | NCR製 53C825を採用．FAST WIDE SCSI-2 対応 | Future Domain 社製．FAST SCSI 対応 | FAST SCSI-2 対応．2048 バイトの FIFO バッファ搭載．レジューム機能搭載，活線挿抜対応 | FAST SCSI-2 対応．144 バイトの FIFO バッファ内蔵．低消費電流設計(MAX 80 mA)，活線挿抜対応 |
| 価格 | 59,800/69,600 | 16,800 | 19,800 | 21,800 |
| 問い合わせ先 | ㈱テクサインターナショナルコーポレーション ☎ 045-935-2560 | ㈱アイ・オー・データ機器 新製品窓口 ☎ 0762-60-1024 | メルコインフォメーションセンター ☎ 052-619-1827 | ㈱アイ・オー・データ機器 新製品窓口 ☎ 0762-60-1024 |

## アップルコンピュータ, Macintosh 漢字Talk 7.5.1
### Macintosh用OSの最新バージョン

今年1月に発表された漢字Talk 7.5を総合的にチューンアップし, 操作性, 処理速度, 信頼性を向上させたほか, MacTCP, QuickDraw-GXをバージョンアップ, また特別付録にDirectDialupMail日本語版の追加により, 遠隔地から電話回線を通じて電子メールの送受信が行えるようになった. アップグレードキットはCD版：24,000円, FD版：27,000円

▶アップルコンピュータ㈱アップルカスタマーアシスタンスセンター ☎0120-61-5800

## ロータス, スーパーオフィスR3.1J
### Windows対応のビジネスソフト5製品セットの新バージョン

各ソフト間の連携機能強化を図り, チームコンピューティング環境の提供を目的としている. 価格は98,000円. R3J登録ユーザーは10,000円でバージョンアップできるほかロータス10周年記念キャンペーン価格(30,000円)を用意. ライセンスパック, アカデミックパックも同時に発売する. 1パッケージ/10ライセンスそれぞれ47,000/400,000円, 40,000/230,000円.

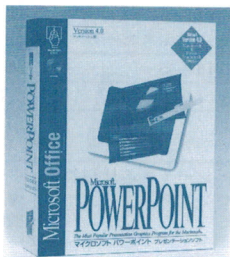
## ロータス, 1-2-3 R2.5J
### DOS版表計算ソフトの最新バージョン

DOS 6.0への対応, パフォーマンスを高めながらコアサイズを縮小, 1パッケージ化によるインストールの改良など, ユーザー調査の結果を基に, 200項目以上の改良が加えられている. 通常パッケージは58,000円, バージョンアップ価格は10,000円. ライセンスパック, アカデミックパック, パスポートプログラムも用意.

▶ロータス㈱インフォメーションセンター ☎03-5496-3111

## マイクロソフト, PowerPoint for the Macintosh Version 4.0
### ビジネスプレゼンテーションソフト

今回のMacintosh版の発売によりWindows, Macintoshの両プラットフォームが混在するオフィスにおいても幅広いデータの共有が可能になる. 従来の680x0系機種, PowerMacintoshのいずれにも対応. 標準価格58,000円, 発売記念キャンペーン：25,000円. アカデミックパッケージ：19,800円, ライセンスパック 追加1ユーザ用/20ユーザ用：46,400/870,000円.

▶マイクロソフト㈱インフォメーションセンター ☎03-5454-2300

## 日本オラクル, Oracle7 Workgroup Server for WindowsNT Version 7.1 Oracle7 Workgroup Server for NetWare Version 7.1

Oracle7 Workgroup Server(OWS)のWindowsNT 3.5およびNetWare 4.1J, 3.12J対応版. 新たにTCP/IPをサポートするとともに, NTクライアントのサポートも追加. WindowsNT上での使いやすさが向上した. また, Oracle Objects for OLEを標準で装備しているので, VisualBasicやVisualC++, OLE 2.0に対応したWindows用アプリケーションからOWSにアクセスが可能. 2ユーザー/5クライアント・パック：79,000円. 5/10パック：190,000円.

▶日本オラクル㈱C/S事業部 ☎03-5213-6334, FAX03-5213-6120

## 富士通, ImageOFFICE for V1.2
### ドキュメント・イメージングソフト

Windowsパソコンで手書き文書などいろいろなイメージデータを自在に管理/活用できるソフト(3,000円). イメージスキャナを接続して個人や部門でのファイリングシステムとして活用するための「ImageOFFICE Personal オプション/Expert オプションV1.2」(30,000/270,000円)も対応スキャナを増やし入力系を強化した.

▶富士通㈱クライアントサーバ推進本部 ☎03-3216-9203

## システムクォリティ, QFILE V6.0
### Windows対応光・カラーファイリングソフト

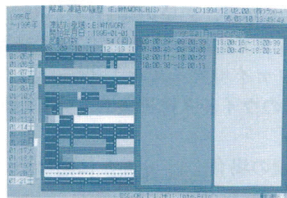
イメージスキャナから読み込んだ一般文書, 新聞記事, 各種図面, カタログなどを各種の検索キーを付けてTIFFフォーマットで光磁気ディスクやハードディスクなどに保存するソフト(128,000円). バージョンアップにより同一キーによる画像の連続保存が可能になったほか, モノクロ(2値, 256階調), カラー(16色, 256色, フルカラー)対応の充実, TWAIN対応スキャナのサポート, QPAINTでのフルカラー画像の編集が可能.

▶システムクォリティ㈱ ☎03-3473-0031, FAX03-3440-0789

### ランハード，Barrier II セキュリティソフト

サブディレクトリまたはルートディレクトリの凍結/解凍を行うセキュリティソフトウェア(18,000円). 安全性には2重3重対策が施されている上，リカバリー機能も用意. そのほかディレクトリの凍結/解凍，パスワードの変更などの履歴が記録されるのでディレクトリの使用状況が一目瞭然となる. 必要に応じて個々の実行ファイルにパスワード機能をつけることもできる. PC98シリーズ，DOS/V機対応.

▶㈱ランハード

☎0559-25-8222

### メガソフト，STARFAX for Windows Ver.4.0 Windows3.1対応 FAX通信ソフトウェア

Ver.3.1の機能を大幅に強化したFAXソフト(15,000円)で，主な特長は次の通り. DDE機能により直接一太郎Ver.6から送信できる，住友電気工業㈱の日本語OCRソフト「OK READER」(98,000円)のサブセット版を標準装備，DOS用の最新FAX通信ソフト「STARFAX Ver. 3.0」(DOS/V，PC98用)を同梱し非ウィンドウズ環境でもFAX通信が可能など.

▶メガソフト㈱サポートセンター

☎06-386-2043，FAX06-386-2123

### システムソフト，キッドデスク日本語版 Windows版子供用デスクトップ

米国エドマーク社製の，コンピュータに不慣れな子供や家族専用のデスクトップ(7,800円). 予め設定してあるアプリケーションのみ表示し，楽しみながらコンピュータに慣れ親しむと同時に基本的な操作や機能を覚えられるユーティリティ.

▶㈱システムソフト インフォメーションセンター ☎092-752-5264

### 富士通，ATLAS for Windows V2.0 英日/日英翻訳ソフト

英日・日英とも13万5千語の基本辞書をもつが，特にパソコン用語5千語を追加しておりパソコンソフトのマニュアルの翻訳に強い. 翻訳の支援機能としてスペルチェック機能，訳文スタイル設定機能，訳語選択機能などを用意. ワープロとの連携として「一太郎 Ver6 for Windows」，「OASYS/Win」，「Word Ver5.0，Ver6.0」の画面から翻訳機能を呼び出すことができる. また文書ファイル名を指定するだけで，DOSテキストファイルまたはOASYS文書を最大10文書まで一括して翻訳できる. 英日・日英とも58,000円(CD-ROM版，FD版).

▶富士通ミドルウェア㈱

☎045-471-0282

### ジャストシステム，読売新聞 検索用 YOMIDAS辞書 1995年度版

読売新聞記事データベース「YOMIDAS」で使用している編集用基本辞書を一太郎用に改良したもので，言葉のグループごとにリードターム(重要語：約12,000語)，同義語(約10,000語)，狭義語，関連語(各約11,000語)などの関連づけがされている. 一太郎 Ver6に搭載されている検索機能での類義語検索の辞書にYOMIDAS辞書を使用すると強力な類義語検索が可能となる. ジャストシステムの登録ユーザ向けメールオーダーサービス，8034円(税込み).

▶㈱ジャストシステム インフォメーションセンター

☎0886-52-8555，☎03-5412-3939

### メディア・ラボ，電辞窓 旺文社国語辞典

中型国語辞典としては類書中最大の7万語を越える見出し語を収録した「旺文社国語辞典(第八版)」のデータベースを基にパソコン上で利用できる電子辞書にしたものでMacintosh版とWindows版がある(12,800円). 主な特長は同音語・同意語の使い分けが解説されていること，豊富な類語が収録されていること，明快な語釈・的確な用例，作業中の文章をそのまま取り込んで検索できること，ヒストリーボタンをクリックすると検索した語句が一覧表ダイアログに表示されることなど.

▶㈱メディア・ラボ

☎03-3225-5371

### パーソナルメディア，名刺男(メイシマン) Windows対応 名刺管理ソフトウェア

シャープの名刺読み取り機と接続して名刺を自動的に読みとり，簡単に検索，並べ替え，ラベル印刷が可能なソフト(20,000円). 電子手帳ザウルスの名刺管理と相互にファイル交換ができる. またペイントブラシの絵を貼り込み，画像入りのデータ管理もできる.

▶パーソナルメディア㈱

FAX03-6702-7850，☎03-5702-7857

### ダイナラブ，DynaFont プレミアム17書体パック PSプリンタ版/他

Mac対応のPSフォント. 製品は，「DynaFontプレミアム17書体パック PSプリンタ版 高解像度用(1201 dpi以上)」(168,000円)「DynaFontプレミアム17書体パック PSプリンタ版 低解像度用(1201 dpi以上)」(68,000円)「DynaFontプレミアム17書体パック PSプリンタ版 ATM版」(12,800円)の計3製品. 平成明朝，平成ゴシック，行書などの書体を含む. 動作環境は，(8Mバイト以上のメモリ，漢字Talk 7以降のシステム)

▶ダイナラブジャパン㈱

☎03-5453-8332

# ウイルス・レポート 51
《Tamanna》

ワクチン入手先
〒105 東京都港区虎ノ門 3-7-2 ☎ 03-3435-1348
〒420 静岡県静岡市常磐町 3-6-11 ☎ 054-252-0085
株式会社 ジェード

このウイルスはメモリに常駐し，COM 型実行ファイルと EXE 型実行ファイルに感染する Tamanna というファイルウイルスです．

このウイルスに感染しているファイルを実行すると，ウイルスがメモリの上限に常駐し，ファイルの実行を監視します．常駐チェックを行っているので，重複して常駐することはありません．

このウイルスがメモリに常駐している状態で，COM 型，EXE 型の実行ファイルを実行すると，実行したファイルにウイルスが感染し広がっていきます．感染したファイルはファイルサイズが 1,857～1,872 バイト増加し，ファイルの属性に読み出し専用属性が付加されます．感染チェックを行っているので重複して感染することはありません．

このウイルスがメモリに常駐してから約 15 分経過するごとに発病し，画面中央に以下のメッセージを表示して T キーの入力を要求します．

```
Wish . . .
Facts need no recognition

RM - A silent friend
DHAKA

Press T to reconcile with my Wish
```

| ウイルス名称 | Tamanna | | |
|---|---|---|---|
| ウイルス俗称 | 不明 | | |
| 起源 | 不明 | | |
| 国内発見日時 | 1995 年 6 月関東地方 | | |
| ステータス | NEC | IBM | 富士通 |
| ▶感染力 | ■■■□□ | ■■■□□ | ■■■□□ |
| ▶露見しにくさ | ■■■□□ | ■■□□□ | ■■■□□ |
| ▶被害 | ■■□□□ | (小)■■□□□(大) | ■■□□□ |
| タイプ | ▶LMB 常駐型<br>▶実行時感染型<br>▶COM 型ファイル後部寄生型(COMMAND.COM を含む)<br>▶EXE 型ファイル後部寄生型<br>▶暗号型 | | |
| ウイルス長 | ▶メモリ常駐 ……………1,872 バイト<br>▶COM 型実行ファイル …1,857～1,872 バイト増加<br>▶EXE 型実行ファイル …1,857～1,872 バイト増加 | | |
| OS | MS-DOS, IBM-DOS | | |
| 症状 | IBM およびその互換機の場合，ウイルスが常駐してから約 15 分ごとに発病して，メッセージ(本文参照)を表示し，T キーの入力を要求する． | | |

T キーが入力されると，以下のメッセージを表示します．

```
OK carry on >>
```

ウイルスは，感染ファイル実行時に自分自身のファイルの先頭を調査し，自分自身で保存しているファイルの先頭と異なった場合，ファイルの先頭を保存しているものに書き換え，以下のメッセージを表示してリブートします．

```
File has been modified
Rebuilding . . .

Standby while rebooting
```

ウイルスは暗号化されているため，デバッガなどで発病文字列を確認することはできません．

## ● 確認方法

感染の疑いのあるファイルを，感染の疑いのないマスタディスクに保存してあるファイルと比較して，COM 型，EXE 型ファイル共に，ファイルサイズが 1,857～1,872 バイト増加していれば，このウイルスに感染している可能性があります．

IBM およびその互換機の場合，15 分ごとに前出のメッセージが画面上に表示された場合，このウイルスに感染していると判断できます．

## ● 復旧方法

まず，感染の疑いのないライトプロテクトされたシステムディスクから起動します．

次に，感染ファイルの読み出し専用属性を解除し，感染ファイルを DEL コマンドで削除して，マスタからの再インストール，またはバックアップからのリストアで復旧します．

## ● 追加パラメータファイル

スキャンワクチンでは，Ver.4 から「追加パラメータ」によるウイルスの発見ができるようになっています．以下のリストは Tamanna の追加パラメータファイルです．エディタなどで編集し，スキャンワクチンと同じディレクトリに VAKZIN.APP の名前でセーブして利用してください．

```
NAME  Tamanna
DATA  0074H, 8AH, 26H, 35H, 00H, FCH, ACH, 32H, C4H, AAH, E2H, FAH
DATA  03FBH, B8H, 1CH, 25H, BAH, 1FH, 01H, CDH, 21H, B8H, 21H, 25H, BAH,
      85H, 00H, CDH, 21H
DATA  0192H, B8H, 23H, 25H, BAH, 14H, 01H, CDH, 21H, BAH, 16H, 01H, B8H,
      24H, 25H, CDH, 21H
DATA  02A5H, B4H, 40H, B9H, 41H, 07H, 90H, BAH, 00H, 00H, CDH, 21H
RAN   T, 0H
COM   L, -1857
EXE   L, -1857
```

# 読者の声

**〈7月号の特集について〉**

■ 今月号の特集は，“なぜ今のDOS/Vがあるのか”などの歴史について書かれており，若い(?)私にとってもわかりやすい内容で興味深いものでした．少しずつ勉強し，早く貴誌のレベルに追いつくようがんばりたいと思います．

▶ でんぷん

■ 16ビットだ32ビットだと高性能化に拍車がかかる昨今だが，20年くらい前にミニコンを勉強しながら4004やZ80でハードウェアを作り，ローダーやモニターのソフトウェアも自作したのがなつかしく思えた．ちょっとした自作も半年から1年の時間がかかり，今は楽になったものだと思う．

▶ 牧田一雄

■ コンピュータ史というと，エッカート・モークリーの時代からいきなりApple IIまで飛んでしまう記事が多いので，今回の特集は貴重だった．こういう，技術者の情熱と遊び心が伝わる記事をこれからも扱ってほしい．

▶ 吉田　拓

■ 15年ほどコンピュータのソフト業界に身を置いているが，ミニコン/パソコンの変遷史には詳しくはない．今号の特集は，技術動向をみきわめるために役に立つ．ハードウェアの購入を検討している現在，歴史を知ることは必要かもしれない．よい企画だ．

▶ KENZO

■ 今月号は久しぶりの×(バツ)でした．トラコンにふさわしくない内容だと思います．買ってしまってから後悔です．来月号はWindows用デバドラで今月分をとりかえしたいと思います．パソコンの歴史なんか知りたくもありません．では来月号にあいましょう．Delphi連載まってます．

▶ M.H

■ 今回の特集の意図がよくわからない．「特集について」のところに書いてあることがどれほどの意味を持つものなのか理解に苦しむ．まあともあれなつかしいマシンの名前がたくさん出てきて楽しかった．

Object PALの記事はよかった．ぜひ連載もして欲しい．このようなものを取りあげてくれるのはトラコンくらいなものだからネ．

▶ 山本　武

■ 今月号の特集にはやや失望．正直言ってパソコン変遷史には興味もないし，今後の役に立つとも思えない．もう少し後から読んでも役に立つような技術を絞った特集を読みたい．

▶ オスギ

■ 今回の特集はあの「電子立国日本」をほうふつさせるような内容で，非常に面白かったです．やはり何ごとも歴史を知るということは大切なことだなと思います．

▶ きくりん

■ この時期になぜコンピュータ史なのか，理解しがたい．「技術を理解」にしては内容が浅くないか？ 確かに「パソコン史」はまとまった読み物はない．が，特集を組むほどのものだろうか？(ミニコン史は面白かった．DGはどうした？)

▶ a.toshi

■ ちょうど大学でPC9801の価格対性能化の推移について調査していたので，今月号の特集はとてもためになりました．ついでにDECやIBMのこともわかって実に得した気分です．

▶ KIYO

■ 今月は買う必要はないだろうと思って本誌を手に取ると，特集に目を奪われてまた買ってしまいました．当方は若い人達に情報処理機器のしくみを教える機会を与えられているのですが，単行本には現状を伝えるものがなく本誌の特集がとても役に立っています．「窓を開けるとリンゴ以外の何が見えるか！」などという特集はいかがでしょうか．

▶ しよしまる

■ 今月号の特集は，自分の今までの仕事の歴史を見ているようでなつかしく振り返って読んでみました．ミニコンを使った自動倉庫システムも今ではパソコンレベルでシステムを作れるようになり，8ビットだった石も今では32,64ビットとなりコンピュータの世界の進歩は早いと思います．後ろを振り返りながらも取り残されないよう頑張ります．

▶ BC+?

■ 7月号の特集は(ひさびさに)すごくよかった(というか，私にも関係があったというか…)．が，その他の記事は，Win関連ばかりで，ほとんど使えるものがなかった．…時代がWinに向かっているから，仕方はないといえばそうなのだが…．しかし，カナシミがとまらない…．うう．

▶ 字がキタナイ

■ 昨年大学でいじっていた機体がPDP-11だった．マニュアルなどを見つつやってたが，知らない規格であるUnibusや言語などが出てきて，教官に聞いてもよく知らないとのことだった．今回の特集でやっと当時の製品概要を理解し得てうれしい．

▶ Siro

■ 「知っておきたいミニコン/パソコン変遷史」を読みましたが，私が疑問に思っていることに関しては記述されていませんでした．私以外にも疑問に思っている人はいると思います．疑問点：「なぜデスクトップのパソコンの筐体はホワイト系で，ノートタイプのパソコンの筐体はダーク系(ブラック)の色なのでしょうか？

▶ 室井英久

■ ちょっと前まで286が最速，なんて思っていたら，P6だのパワーPCだのって，すごいですね．これは4ビットや8ビットのソフト開発をする私にとっても，アセンブル時間の短縮というありがたい影響を与えてくれました．ところでパイプラインとか使っているCPUで1マシンサイクルの時間て正確にわかるんでしょうか？

▶ 4 bitの心

**〈その他のご意見〉**

■ 5月号のAT組み立て記事に対して，軟弱だとか，

# Readers' Square

3日で読み切れるほど簡単だったとかいう読者から反応がありますが，私はあれで良かったと思う．何しろ資料/リファレンスとしてしっかりしているし，私はそのためにトラコンを買っているのだから．でもマゾな読者も多いなと感心した．

▶徳田献一

■「From U.S.A. ①英語の発音は難しい」(7月号 p.139)を読んで，「そのとおり！そのとおり！」と頷く私ですが，開発の現場で良く使われる大きな大間違い(発音について)をちょっと紹介します．length [lέŋ(k)θ] なのに「れんぐす」と呼ぶし，width [widθ] は「わいず」と読んだり，極めつけはMacのフォント名でGeneva [dʒiníːva] を「ジェノバ」と読んでイタリアの都市名にしていたりするんですよ．

▶牧　和寛

■ Windows Programming FAQ についてですが，Visual Basic から Windows API を使う場合参考になります！ Visual C++だけでなく，Visual Basic で利用する場合での解説もあればもっといいのに…!!

▶立橋良彦

■ LISP の記事を読んで，LISP のソフトを買いました(TUTScheme)．LISP ができると OS などのシステムプログラミングのセンスが身につくと聞いて使ってみたくなった．今まで BASIC，C，Fortran，MASM を使っていましたが，それらの説明のシンボルが LISP を使っていることがわかって学習したくなりました．

▶ J. TAKADA

■ MS ACCESS のプログラミング特集をお願いします．Visual Basic の記事はありますが，ACCESS はまだです．書店にも入門シリーズのみで，適当なものがありません．

▶蔵衛門

■ 普段，Excel を使っていても，簡単なグラフと計算しかやらないで，ほとんどマクロなどつかわないけど….数回の VBA の連載を読んでいて，何か興味がわいてきて，ちょっと勉強を始めました．これからも，具体的な例をたくさんのせてがんばってください．

▶服部彰人

■ 今年に入ってからの「Excel 5.0 の VBA 活用マニュアル」連載を毎号期待しています．VBA で自動化をするチョッとしたことがあったのですが，15 インチ(私の会社での PC)で作ったものが，17 インチディスプレイの PC 上で動かすと，ページ当たりの行数が狂

## 投稿募集のお願い

本誌では，読者の皆さんが作成されたプログラムなどを募集しています．もし，プログラムなどを発表したいと思われたなら，是非とも本誌編集部までご連絡ください．

お手紙でも，電話(03-5395-2124)でも，NIFTY-Serveを使ったメール(ID番号：HBG01366)でも，何でもOKです．

## 本誌掲載プログラムを利用される方へのお願い



## CONGRATULATION

モニタープレゼント(7月号)当選者

① まいと～く Ver.3［㈱インターソフト 提供］
東山梨郡・河埜和人

読者プレゼント(7月号)当選者

Ⓐ Compaq 社特製アナログ時計
尼崎市・宮部敬司

Ⓑ NetImage 社特製ポロシャツ
府中市・荒井信明

CD-ROM プレゼント(7月号)当選者

★ OS/2 mate
国分寺市・山木光人

★ Blowfish OS/2
習志野市・稲垣勝良

## 訂正

● **1995年8月号**

p.62 左↑5 **0000～0000：007F** の割り込みベクタの領域

p.64 **〈図4-7(b)〉** 内「同じページテーブルを示している」の下側の矢印を朱色網かけ部に移動

p.69 **〈図5-1〉** 内 **Win386_Startup_info_Struc**(SIS)
**SIS_Reference_Data**
［バイト数］**+2** → ［バイト数］**+4**
［ES：BX］の矢印をアドレス **+2** の
［セグメント｜オフセット］へ移動．
［次の SIS］の矢印の始点をアドレス **+6** の
［セグメント｜オフセット］へ移動．

p.85 左↑7 ● **6h：sendblk**〔ブロック送信〕

p.86 **〈図3-2〉** 内 (a) 送信レジスタ空割り込み処理

p.88 右↓1 ●モデムステータス割り込み処理

p.94 右↑8 G.W.Johnson

p.96 右↓3 scroll

p.99 ◆参考文献◆内 (6)，(9) G.W.Johnson

p.109 右↑2 **hs_DiagCross**

p.150 ◆引用文献◆

(1) DT Network System Guide Version 4.4, InterSystems Corporation (1993)

い，プリント時にズレて出力されてしまいます．原因は何でしょうか？
▶山田伸明

▌HPFS や NTFS の内部構造を知りたいのですが，解説をお願い致します．

それと，今月号は「ちょっと息抜き」という感じでした．でも連載がなかったら買わなかったかも？
▶鹿田勝行

▌三つの O が付くソフトが気になっています．それは，Oberon，Occam，OS-9(000) で，マルチメディア時代においてこれらの実力はジャストフィットすると思うのですが，世間では顔の良い(GUI)ソフトしかもてはやされていない．さみしい限りです…．▶都倉有美

▌DOS/V の PRN が紙切れなどの検出を行うのは当たり前だと思っていました．PC-DOS では紙切れ検出はしないんですか？ でも，この程度のことならユーティリティを作れば簡単に対処できますよ!! 10 行プログラムにちょうど良いテーマじゃないですか．
▶麻由美

▌「GNU C/C++ Compiler でマスターする OS/2 プログラミング」を毎号楽しみにしています．そこで質問ですが，GNU C/C++ Compiler にはデバッガというのはあるのでしょうか？ もし存在するならデバッグの方法も解説をお願いします．▶issy

▌記事の掲載方法について一言．文章と図表の関連をあまり考慮せず，ただ紙面の空いているスペースに詰め込んでいるように思えます．文章とそれに関連する図・表を同一ページに，できれば，横にならべて掲載してください．
▶和栗輝久雄

▌日本の雑誌で見かけることはめったにない Amiga．ドイツで製造が再開されるそうですが，ああいう面白いマシンにはもっと普及してほしい．日本に入ってきたら買うかも(A500 ユーザー．あ，今使ってないや)．
▶BIMBO

▌貴社発行のエレクトロニクス関係の雑誌をいろいろと購入しました．中でも「アーカイブ」という，ちょっとカタメのコンピュータ関係の本が気に入りました．いつの間にか消えましたね！ トラ技コンピュータもそうなってしまうのかと気になってしかたがありません．がんばってください．
▶その他 1 名

▌とてもなつかしかったです．

誰か教えてください．お願いします．FAXMODEM からファックスへ送信するテクニックを!! またその逆も．ファックス側でエラーが出て送信側チェックをと表示される．G3 に対応できるとマニュアルに書いてあるが？
▶Jara

## 通信販売の送料変更について

8 月 1 日(火)当社到着分の通信販売より，送料をご購入金額別の送料荷造り費に変更させていただきますので，あしからずご了承願います．詳細につきましては，緑の頁「出版案内 14 頁」に記載してありますので，ご覧ください．

## 年間予約購読を受け付けています

トラ技コンピュータの年間予約購読ができます．ご希望の方は，CQ 出版社 販売部まで，または本誌のアンケートはがきを使ってお申し込みください．折り返し振り替え用紙をお送りします．購読料は送料，消費税込みで 1 年間10,320円です．

## バックナンバのお知らせ

トラ技コンピュータのバックナンバが入手できます．ご希望の方は，お近くの書店あるいは CQ 出版社販売部(住所はディスク頒布の申し込み先と同じです)までお申し込みください．

## ディスク頒布のお知らせ (今回は 1995 年 7 月号と 8 月号，9 月号の内容)

1995 年 9 月号の内容はニフティサーブの FDEVICE フォーラムにアップしてあります．

本誌 1995 年 9 月号で掲載したプログラムのソースリストをフロッピで頒布します．ただし，市販の言語処理ソースなどにはすべて著作権がありますので一緒に頒布できません．コンパイラなどの処理系は読者自身がご用意ください．また，ソースリストも読者の入力の手間を省くことが主な目的ですので，プログラム上のバグや本誌に掲載した内容以上の質問には応じられないことがありますのでご了承ください．同ディスクには 1995 年 7 月号および 1995 年 8 月号の掲載プログラム，トラコンInformation およびトラコン・データベースの最新版も入っています．ご希望の方は現金書留か定額小為替で代金 3,000 円(送料・税込み)と下記の申し込み書のコピー(手書き，ワープロでも可)を CQ 出版㈱デザインウェーブまでお送りください．なお，頒布メディアは MS-DOS フォーマットの 3.5/5 インチ 2HDです〔申し込み書にどちらかをチェックしてください．指定がない場合は 3.5インチ2HD(PC98 用)をお送りします〕．

なお，今月号のディスクサービスの受け付け期限は 1996 年 1 月末とします．

**〈送り先〉**

〒170
東京都豊島区巣鴨 1-14-2
CQ 出版㈱
FD サービス係 行

◆申し込み書 TC9509 〈トラ技コンピュータ 1995 年 9 月号 ディスク・サービス〉

| | | |
|---|---|---|
| PC98 | 3.5□ | 5□ |
| IBM | 3.5□ | 5□ |

ご住所 〒　　　－

お名前　　　　☎(　　)　　－

会員番号* CQ□□□□□□□□□

＊資料請求で会員に登録されている方は会員番号をご記入ください．

コンピューティング・エッセイ 36

# TV ゲームをあなどるなかれ

北海道大学　山本　強

昨年末から本格化した32ビットゲームマシンの開発競争も，ソニーのPlayStation，セガのサターンともにわずか半年あまりで100万台規模の市場を作ったというから，この不景気な世の中にあって，数少ない景気のよい話になっている．

私の研究室でも，これらが発売されたその日から学生諸君が持ち込んで「評価」を行っていた．なるほど，確かに凄い．やはり，専用グラフィクスエンジンを搭載した専用アーキテクチャだからと言われると，妙に納得させられてしまうのである．さすがに，普通のPCではこうは行くまいと思っていた．

ところが，世の中にはまだまだ凄いソフトがあるもので，普通のIBM PCでフルグラフィクス，それも毎秒30コマで動いてしまうフライトシミュレータや完全テクスチャマッピングされた3次元戦闘ゲームなんていうものもあるという．

それでも，ゲーム事情通の学生に言わせると，それは普通のことだとあっさり言われてしまうが，そう言われると世のコンピュータグラフィクスの専門家諸子はちょっと困ってしまうのである．

だって，バーチャルリアリティだの，リアルタイムシミュレーションだのといって，莫大な予算を使っている理由の一つに，PCや通常のワークステーションではリアルタイムグラフィクスが実現できない，だからグラフィクススーパーコンピュータ5千万円也が必要だということになっているのだから．

とはいっても，別にPCがたいしたものではなくて，リアリティエンジンだから凄いという訳ではなく，最終的に画像として見えたときにどれだけの差があるのかということに尽きるはずである．

それはそれでよい．しかし，どうしてたかがPCで，昔はグラフィクスエンジンがなければできなかったような高速レンダリングができるようになってしまったのか，という素朴な疑問は全く解決していない．

去年，DOOMという戦闘ゲームのグラフィクスでかなりインパクトを受けたが，まあ，あれくらい手を抜けばできないこともないかなと考えていた．とはいうものの自分でそれ級のソフトを作ってみたわけではないので，できるかどうかの自信はいま一つであった．

それを忘れかけて1年ほど経ってしまったが，ここにCommanche™ Maximum Overkill™という空中戦シミュレータのゲームソフトがあるのだが，このスペックを見て再び愕然とした．このソフトのレンダリングスペックは，フルテクスチャマッピングで1/30秒当たり48,144要素を処理するという．

ちょっとまて．それなら1秒では144万要素になるではないか．シリコングラフィクスが誇るリアリティエンジンだって100万ポリゴン毎秒なのである．

Voxel Space(pat.pending)という特別なアルゴリズムを使うと説明書には書いてあるが，どうすれば100万要素以上の処理が66 MHzの486 CPU 1個で出来てしまうのか….

私も大学でコンピュータグラフィックスの講義を持っているが，これが144万要素を1秒間に処理できる理由を説明できないのが悲しい．説明できないというのは，いわゆる教科書に解説されているオーソドックスな手法ではできそうもないということで，実際にできているのだから方法はあるはずである．

このゲーム，実際にやっているところを見せてもらったが，多少絵は荒いものの毎秒30フレームを生成できているように見える．もちろん音付きで．ただ，注意深く画面をみると独得の癖のある画像で，まっとうなレンダリングをしていないことだけは確からしい．

ところで，日本が弱いと言われるソフト業界において，ゲーム業界だけは例外的に世界的に高いシェアを誇る分野である．そして，この業界にはいわゆる一流大学を優秀な成績で卒業した学生が「行かない」分野としても知られている．

確かに，ゲーム開発には知識や経験よりもセンスとユニークなアイデアが重要だと言われるから，しっかり教育された人よりも人と違ったキャリアを持っている人のほうがよいと言うのもわかる．しかし，私がいま驚いているのは，このゲームソフトに組み込まれているグラフィクスエンジンのパフォーマンスなのであって，ゲームのアイデアではないのである．

思うに，中途半端にまっとうな理論教育を受けた優等生がリアルタイム3次元ゲームの開発に取り掛かると，まず高速レンダリング，すなわちZバッファあるいはプライオリティ法という図式がでてくる．

そして，与えられたCPUの速度からレンダリング速度を見積もり，これではゲームにならないとなり，開発を断念する．見通しがよすぎるのも考えものである．

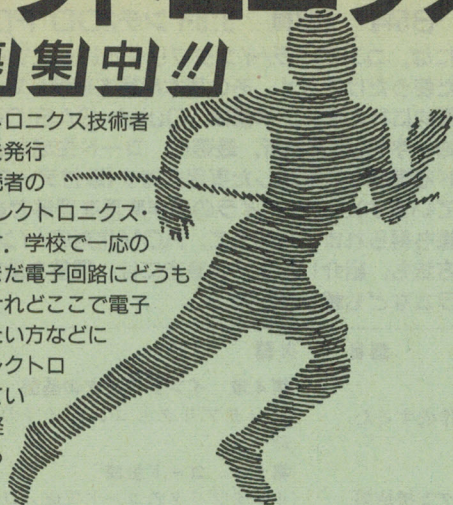
逆に，そんなことを知らないアウトローは既製概念の枠にとらわれないから，ときにはデータやシミュレーション空間に制限を加えて妥協できる点を見つけてしまうのかもしれない．時として，教科書には書かれていない，理論的には欠陥だらけの方法を編み出してくるのではないだろうか．

こうなると，大学の教育，特にCGの教育はどの程度社会の役に立っているのかとても不安になってくる．

わかる 新しい技術力強化プログラム

# エレクトロニクス・セミナー

## 参加者募集中!!

CQ出版社では，エレクトロニクス技術者に役立つ多くの実務書籍を発行していますが，昨年から読者の皆様のご要望に応えて，エレクトロニクス・セミナーを開講しています．学校で一応の勉強はしてきたけれど，まだ電子回路にどうもなじめない方，技術系だけれどここで電子回路の常識をマスターしたい方などにおすすめします．またエレクトロニクスにはほとんど触れていなかったという方にも気軽に参加していただけるよう配慮してあります．

日　程　標準は2日間コース（テーマによっては3日間コースもあります）
時間割　第1時限：am10：00～11：30（昼食）
　　　　第2時限：pm1：00～2：30
　　　　第3時限：pm3：00～4：30
　　　　交流会：1日目のpm4：30～6：00
会　場　専用セミナー会場（JR巣鴨駅，都営地下鉄三田線巣鴨駅から徒歩2分　炭七ビル内）
定　員　42名（定員になり次第締め切らせていただきます）
参加費　38,000円（2日間コースの場合）（テキスト代，食事代，ミーティング代，消費税を含む）

---

著者セミナー　第19回

### 「光デバイスの使い方徹底研究」

──情報化社会で活躍する光素子をマスターする

**開催日**　1995年8月31日(木)～9月1日(金)　2日間　**参加費**　38,000円　**申込締切日**　8月25日
**講　師**　片山　貴雄氏，高橋　望氏（㈱東芝　光半導体応用技術部）

内　容

情報化社会といわれる現代で，光に関連する半導体素子は，光エレクトロニクス技術の中心デバイスとして多くの場面で活躍しています．とくに高信頼度表示技術，データ伝送などに出番がふえています．

このセミナーでは光デバイスの基礎知識から使い方の注意点や応用回路までを，経験豊富なトップ技術者がわかりやすく説明します．

次世代の情報伝送技術，表示技術の中心となる光デバイスの世界に足を踏みいれてみませんか．

①可視光LEDランプと7セグメントLED
②LEDドットマトリックス・ディスプレー
③赤外線受光・発光素子
④フォト・インタラプタ
⑤フォト・カプラとフォト・リレー
⑥光伝送デバイス

---

著者セミナー　第20回

### 「個別半導体デバイス活用法のすべて」

──トランジスタ，FET，ダイオードの実用回路マスター

**開催日**　1995年9月21日(木)～22日(金)　2日間　**参加費**　38,000円　**申込締切日**　9月11日
**講　師**　青木　英彦氏（㈱東芝　半導体システム技術センター　LSI技術第1部）

内　容

電子回路にはIC，LSIが多く使われていますが，その周囲では必ず個別半導体が重要な働きをしています．ICでは実現できない多くの機能（定電圧，整流，発光，保護など）のほか，ICの分担する領域以外での低雑音，高周波，高電力などの分野はまさに個別半導体の独壇場です．

個別半導体をうまく使いこなすと，ICを使った回路の最高性能が発揮できます．また個別半導体の動作を知ることによって，IC，LSIの特性の理解もさらに進みます．このチャンスに個別半導体回路をマスターしましょう．

①いろいろな個別半導体素子とその特徴
②ダイオード1．小信号用（汎用）ダイオード
③ダイオード2．整流用ダイオード
④ダイオード3．定電圧ダイオード，LED
⑤トランジスタ1．小信号（汎用）トランジスタ
⑥トランジスタ2．パワー・トランジスタ
⑦FET

著者セミナー　第21回「エレクトロニクスの世界へどうぞ」
──フレッシャーズのための電子工学講座

開催日　1995年10月5日(木)～6日(金)　2日間　参加費　38,000円　申込締切日　9月28日
講　師　吉本　猛夫氏（㈱ケンウッドエンジニアリング　技師長）

内　容

エレクトロニクスの技術進歩があまりに速いために，エレクトロニクスの基本的な技術を習得するチャンスが減っています．とくに，これからエレクトロニクスの世界へ入っていこうとする新人や，他の分野の人々にとっては，適切なガイドが極めて少なくなっています．エレクトロニクスの全体は見えにくいほど大きいのですが，このセミナーで取り扱う分野を理解すれば，基礎体力が充分身につきます．電気のことはわかっていると思っていても，常識とは違った意外な落とし穴もあります．セミナーではこういう“うっかりミス”しやすい知識もていねいに取り上げます．

①電子回路の基礎の基礎
②トランジスタの基本的な使い方
③IC，LSIの選択から応用まで
④実用回路オンパレード
⑤通信技術と電源の利用
⑥回路部品と電子機器の組み立て

第22回「電源回路設計の基礎」（仮題）

開催日　1995年10月26日(木)～27日(金)　2日間　参加費　38,000円　申込締切日　10月16日
講　師　佐藤　守男氏（大平電子㈱　代表取締役社長）

## エルテスで力だめししてみませんか？

●エルテスとは…ELTES（Electronics Technology Evaluation System）はこのセミナーのために特別に開発された比較式エレクトロニクス技術力測定システムです．

●60問の4択式質問に40分で（レベル1の場合）答えていただき，六つの部門（基礎と部品，半導体，アナログ回路，ディジタル回路，コンピュータ，実装）でのそれぞれの得点をいくつかの業界のトップ技術者の得点と比較します．ご自分がどの分野に強いのか，同業他社の技術者と比べてどうかを客観的に知る良いチャンスです．希望者のみ測定します（2日目の昼食後）．測定結果は，ご自身に，ご希望ならば会社にもお知らせします．

セミナーに参加ご希望の方は下の申込書に記入し，FAXで，あるいはコピーを右記まで郵送してください．

〒170　東京都豊島区巣鴨1-14-2
CQ出版株式会社内
エレクトロニクス・セミナー事務局
TEL:03-5395-2125　FAX.03-5395-1255

## セミナー参加申込書

（申込書1枚に1名分をご記入ください）　（該当する□に✓をしてください）

| 参加者氏名 | | | |
|---|---|---|---|
| 会社・団体名 | | | |
| 所属・部門 | | | |
| 会社・団体所在地 | 〒 | | |
| 電話番号 | | 参加費請求先 | □会社　□本人 |
| ＦＡＸ番号 | | 送金の方法 | □銀行振込　□現金書留 |
| 申込者(参加者と異なるとき) | 氏名 | 所属 | （FAX　　） |
| 参加希望セミナー回数 | 19□　20□　21□　22□ | | |
| エルテス(技術力測定)について | □参加してみたい | □上司に知らせてもよい | |
| 得点を知らせてもよい上司の所属，お名前 | 所属，肩書 | | お名前 |

TRY COMPUTING BOOKS　　8月下旬発売予定

# Cプログラミング書法

岩田憲和
花沢　満 共著
和佐野哲夫

## 移植性の高いプログラムを書くために

予価2,500円
B5変形判
ISBN4-7898-3532-4

本書は，プログラムの開発者が再利用しやすいプログラムを作るにはどうすべきかを理由とともに示した実践書です．すなわち，再利用しやすいプログラムを作成するために，コーディングの前に考慮すべき事項と，再利用しやすいCプログラミング作法を読者に示すことを目的としています．

読者層としては，初級／中級プログラマや学生のテキスト，あるいは補助教材として使用されることを想定しています．また，プログラムの開発を外部に発注するときのハンドブックとしても利用できると思います．読者は，C言語のプログラム作成に慣れていることを想定していますので，経験の深いプログラマにも有益であると確信しています．

〈内容〉 **第1章** プログラムの流通とは／**第2章** Cプログラムの流通を阻むもの／**第3章** 流通しやすいプログラムを作るには／**第4章** ANSI準拠コンパイラ間の流通/K&RとANSI間の流通

---

Cによる実践的プログラミングの手引き

# 必ずわかるC言語

好評発売中　土井滋貴 著
B5変形判　240頁　定価2,900円
ISBN4-7898-3533-2

手軽に，かつスピーディに役立つCプログラムを作成するためのノウハウを養う実践書です．hello worldを卒業し，次なる書を求める初級者にとって，格好のテキストになるように書かれています．たとえば，データ構造やアルゴリズムを学ぶ必要のあるところでも，プログラムの動作を理解することにより，自然にそれらを身につけていくことができるように構成されています．

## 既刊（すべてB5変形判）　好評発売中

やさしいグラフィック・デザインのためのテクニック集
**マック ＆ デザイン**
濱田功志 著　120頁　定価2,800円
ISBN4-7898-3537-5

実行しながら学ぶ
**DOSユーザのためのUNIX**（入門編）
堀内泰輔/不破 泰 共著　中村八東 監修　264頁　3.5″2DDFD付　定価2,800円　ISBN4-7898-3547-2

先進のかな漢字変換をつかいこなすために
**WX II-Win2.1活用法**
石渡紫朗 著　170頁　定価2,100円
ISBN4-7898-3538-3

ANSI Cの標準関数の詳細と使い方
**C言語 実践プログラミング入門**
大貫広幸 著　226頁　5″2HDFD付　定価2,800円
ISBN4-7898-3548-0

実行しながら学ぶ
**DOSユーザのためのUNIX**（システム管理編）
堀内泰輔/不破 泰/佐谷野敏晴 共著　中村八東 監修
264頁　定価2,600円　3.5″2DDFD付　ISBN4-7898-3539-1

技術者のための
**TURBO PASCAL活用法**
中野正次 著　228頁　5″2HDFD付　定価2,700円
ISBN4-7898-3549-9

MS-DOSからOS/2,MS-Windows(Visual Basic) まで対応する
**実用RS-232C通信プログラム作成法**
PC98,FMR,B16,IBM PC/AT対応
横田秀次郎 著　230頁　定価2,500円　ISBN4-7898-3540-5

A-D/D-A変換とパラレル入出力拡張ボードで構築する
**パソコンによる計測・制御の実践入門**
トラ技コンピュータ編集部 編　216頁　定価1,900円
ISBN4-7898-3550-2

表計算ソフトを使った科学技術分野へのアプローチ
**Lotus 1-2-3による理工系シミュレーション入門**
臼田昭司/井上祥史/伊藤 敏 共著　216頁　5″2HD FD付　定価2,900円
ISBN4-7898-3541-3

高校物理・数学の演習から問題解決まで
**パソコンによる数式処理ソフト活用法**
臼田昭司/井上祥史/伊藤 敏 共著　200頁　定価2,400円
ISBN4-7898-3535-9

DSPとリアルタイム信号処理の基礎から応用まで
**ディジタル信号処理プログラミング入門**
三上直樹 著　240頁　定価2,800円
ISBN4-7898-3545-6

Visual C++とBorland C++による
**はじめてのWindowsプログラミング**
沖 充弘 著　198頁　定価2,200円
ISBN4-7898-3536-7

マニュアルを越えたデータベース言語のプログラミング・ノウハウ
**技術者のためのdBASE活用法**
丹羽一夫 著　296頁　3.5″2DD FD付　定価3,000円
ISBN4-7898-3546-4

BIOSからファイル転送，通信プロトコルの作成まで
**パソコン通信プログラムのすべて**
由本正英 著　216頁　定価2,900円
ISBN4-7898-3534-0

CQ出版社 〒170 東京都豊島区巣鴨1-14-2 販売部☎03-5395-2141 振替 00100-7-10665

電子制御ガソリン噴射の基礎から
コンピュータの製作まで

# 電子制御エンジンの基礎・応用

**淺原　宏 著　A5判　160頁　定価2,100円**

ISBN4-7898-3066-7

**重版出来！**

最近の自動車は，エレクトロニクスで走っているといっても過言ではないほど電子制御化がすすんでいます．その核となるのはエンジンの電子制御技術でしょう．本書は，今までベールに包まれていたその技術を，実例を示しながらわかりやすく説明します．また，RS-232-Cインターフェースや学習機能を搭載した独自のエンジン・コントロール・コンピュータを製作し，具体的な制御技術を解説します．

**第１部　電子制御ガソリン噴射の基礎**

第１章　電子制御ガソリン噴射の基本概念／第２章　電子制御ガソリン噴射の種類／第３章　構成部品／第４章　燃料噴射時間の算出／第５章　点火時期制御／第６章　アイドル・スピード・コントロール／第７章　故障に対する備え／第８章　そのほかの制御／第９章　地球に優しい自動車をめざして

**第２部　EFIコンピュータの製作**

第10章　基本構想／第11章　ハードウェア／第12章　ソフトウェア／第13章　セッティング／第14章　あれこれ作ったもの(アナログ式サブコン，エンジン・パワー測定器など)／資料集：各社クランク角センサの出力波形

---

# 高周波回路の設計・製作

鈴木憲次 著　A5判 248頁
定価 2,100円
ISBN4-7898-3065-9

**●回路設計の基礎から実用回路の設計まで**

本書はトランジスタ技術で'90年5月号から連載された「実験で学ぶ高周波回路」を，単行本用に全面的に書き改めたものです．高周波回路を身に付けるには，頭の中のトレーニングだけではだめで，自作の経験度が一番ものをいいます．この本では，各種受信機回路から高周波専用の測定グッズまでを手作りしていきます．基板の頒布サービスもあります．

# ディジタル回路の設計・製作

湯山俊夫 著　A5判 292頁
定価 2,200円
ISBN4-7898-3064-0

**●製作の基本からホビー・実用回路の設計まで**

本書は，ディジタル回路の設計と製作を基礎から解説したディジタル回路の入門書です．前半の基礎編においてディジタル回路製作の基本を詳細に解説しています．また後半の製作編ではまず平易なディジタル回路の製作を行い，しだいに本格的な回路設計に入っていきます．

# ディジタル回路テイクオフ指南

小林芳直 著　A5判 256頁
定価 1,700円
ISBN4-7898-3062-4

**●回路設計を越えたセンスを身につけよう**

最近はノイズ対策や実装技術を身につけないと確実なディジタル回路を構成できない時代になってきました．本書はそんなノウハウといわれている部分をやさしく解説しています．

# アナログ回路の設計・製作

青木英彦 著　A5判 248頁
定価 1,700円
ISBN4-7898-3061-6

**●現実的な回路の作り方と実際の設計法**

本書はこれからアナログ回路を学ぼうとする人たちの入門書です．基礎編では，製作技術やOPアンプ，トランジスタ，ダイオードなどの使い方を紹介し，製作編では，パワー・アンプ，電源回路などを製作しながらその設計課程を詳しく解説していきます．

# センサ応用回路の設計・製作

松井邦彦 著　A5判 256頁
定価 1,700円
ISBN4-7898-3063-2

**●実戦のための応用ノウハウを身につけよう**

本書では“作って学べる本”をめざしました．したがって，各センサについて，センサの概要を説明したのち，つぎにもっとも基本的な回路を紹介して，最後に実用的な回路を設計するような構成にしました．

CQ出版社　〒170 東京都豊島区巣鴨1-14-2　販売部☎03-5395-2141　振替00100-7-10665

別冊 インターフェース　好評発売中

# パソコンによるISDN通信

## ISDN通信回線の利用と推奨版APIによるプログラミング

多賀 直久〈(財)テレコム高度利用推進センター・監修〉 B5判 204頁 定価2,100円

今やパソコンもネットワークに接続されるのがあたりまえの時代になりつつあります．ネットワークといえばLANを思い浮かべますが，LANの敷設されていない遠隔地とのデータ転送をしたいことも少なくありません．また，インターネットと結びたい人も少なくないでしょう．このとき，一般公衆回線＋モデムよりも高速伝送が可能な「ISDN公衆回線」を使用すれば，64kbpsでディジタル・データ伝送が行えます．すでに，ISDN公衆回線は，全国のほとんどの主要都市においてサービス運用がされており，その料金体系も，従来の電話用公衆回線とあまり変わりません．

本書は，パソコンによるISDN通信を行うためのハンドブックです．ISDNの基本を示したあと，ISDN通信に必要な機器やソフトを紹介しながら，ISDNを使用したシステム構築について，実例をあげながら解説しています．とくに，市販されているパソコン用ISDNソフトのほとんどは，FRAPI-AというAPI(ソフト・インターフェース）を使用していますが，このFRAPI-Aについてもくわしく解説しています．

## IFセレクション　好評発売中

### MS-DOSディスク管理技法

中島信行 著 B5判 208頁 5″2HDFD付 定価2,500円

**ブロック・デバイス・ドライバ作成からディスク管理メカニズムの解析まで**

ISBN4-7898-3515-4

MS-DOSの未公開ファンクション・コールの分析をしたあと，その機能を活用してディスク用のデバイス・ドライバの作成法を探り，後半で，DOSのディスク管理のメカニズムの解析とその関連ユーティリティを作成します．紹介したユーティリティの全ソースと実行形式ファイルを収納したFD付き．

### Cによるメモリ管理技法

中島信行 著 B5判 224頁 5″2HDFD付 定価2,500円

**MS-DOS上でTSRプログラム＆デバイス・ドライバを作成する**

ISBN4-7898-3487-5

本書では，常駐プログラムをCで書くノウハウを徹底的に解説しました．また，実例として実用ユーティリティのソースも多数示しています．(MS-C5.1/6.0，TurboC2.0，TurboC++，一部LSI　C86　ver3.20に対応)

### MS-DOS完全活用法

中島信行 著 B5判 224頁 5″2HDFD付 定価2,500円

**DOS起動原理からユーティリティ作成まで**

ISBN4-7898-3485-9

まず，MS-DOSやプログラムがどのようなメカニズムで起動するかを探り，つぎに各システム・ファイルの機能を解析し，アセンブラやC言語を使って，機能活用のユーティリティを多数作成します．さらに，パソコンが用意するROM　BIOSについて，その利用法を研究します．

### オブジェクト指向のすべて

小暮裕明 著 B5判 176頁 1,650円

**'90年代プログラマの必須知識OOPSをマスタする**

ISBN4-7898-3482-4

オブジェクト指向は，細かい規則ではなく，大きな枠組み（パラダイム）で，そのセンスを感じとるには，オブジェクト指向をとりまく話題や実例にできるだけ多く接することが近道です．そこで本書では，ここ数年に著者が体験し，見聞きして得たオブジェクト指向の成果を，あますところなく紹介しています．

### MS-DOSメモリ管理ソフト技法

中島信行 著 B5判 224頁 定価1,650円

**メモリ常駐ソフト＆拡張メモリ活用プログラミング**

ISBN4-7898-3484-0

MS-DOSを使いこなそうという中級プログラマのために，MS-DOSプログラミングのうち，メモリ管理関連手法に的を絞って，徹底的に解説しました．また多数掲載した各プログラムは，どれもMS-DOSを活用するための実用ユーティリティなので，読み進むうちに，プロのノウハウが自然と身に付きます．

### MS-DOSプログラマーズ・バイブル

阿部英志 著 B5判 384頁 定価2,500円

**内部構造とシステム・コールとデバイス・ドライバと…**

ISBN4-7898-3481-6

C言語とアセンブリ言語を用いた応用プログラムの作成に必須となる技術情報を，システム・コールの使いかたとデバイス・ドライバの作かたを中心に，MS-DOS　ver3.30に基づいて具体的に解説しました．

### Visual Basic for Windowsの本格的応用

インターフェース編集部 著 B5判 144頁 3.5″2DDFD付き定価1,800円

**Windows DLLの利用からカスタム・コントロールの作成まで**

ISBN4-7898-3369-0

Visual BASICによるWindowsアプリケーションの作成法として，DLL内の関数を利用する方法，VBXを利用する方法，シリアル・ポート(RS-232-C)を利用するためのVBXの作成方法について解説しました．全ソース＆実行ファイルが，付属のフロッピ・ディスクに収録されています．

### 割り込みプログラミング技法

中島信行 著 B5判 208頁 3.5″2DDFD付 定価2,400円

**パソコンにおける割り込みメカニズムからデバイス・ドライバ作成まで**

本書では,タイマ割り込み(8253,8254)，VSYNC割り込み，プリンタ割り込み，RS-232-C割り込み(8250B，8251A)に関して解説を行い，簡単なプログラムを作成します．パソコンで実際にプログラムを試しながら，ハードウェア割り込みを中心とした割り込みの活用法を探っていけるような構成です．

CQ出版社 〒170 東京都豊島区巣鴨1-14-2 販売部☎(03)5395-2141 振替 00100-7-10665

エレクトロニクスの基礎と実用技術を濃縮したフィールド・ワーク・マガジン

# トランジスタ技術SPECIAL

季刊●B5判●定価：①～㉝定価1,540円 ㉞～㊺定価1,600円 ㊻～㊾定価1,690円 ㊿以降定価1,800円

**①個別半導体素子 活用法のすべて**
基礎からマスタするダイオード，トランジスタ，FETの実用回路技術

**③PC9801と拡張インターフェースのすべて**
16ビット・パソコンを使いこなすためのハード&ソフト

**④C-MOS標準ロジックIC活用マニュアル**
実験で学ぶ4000B/4500B/74HCファミリ

**⑤画像処理回路技術のすべて**
カメラとビデオ回路，パソコンと隔合させる

**⑥Z80ソフト&ハードのすべて**
基礎からマクロ命令を使いこなすまでのノウハウを集大成

**⑧データ通信技術のすべて**
シリアル・インターフェースの基礎からモデムの設計法まで

**⑨パソコン周辺機器インターフェース詳解**
セントロニクス/RS-232C/GPIB/SCSIを理解するために

**⑩IBM PC&80286のすべて**
世界の標準パソコンとマルチタスクの基礎を理解する

**⑪フロッピ・ディスク・インターフェースのすべて**
需要の急増するFDDシステムの基礎から応用まで

**⑫入門ハードウェア 手作り測定器のすすめ**
電子回路設計の基礎と実践へのアプローチ〈在庫僅少〉

**⑬シミュレータによる電子回路理論入門**
コンピュータを使ったアナログ回路設計の手法を理解するために

**⑭技術者のためのCプログラミング入門**
MS-C，Quick C，Turbo Cによるソフトウェア設計のすべて

**⑮アナログ回路技術の基礎と応用**
計測回路技術のグレードアップをめざして

**⑯A-D/D-A変換回路技術のすべて**
アナログとディジタルを結ぶ最新回路設計ノウハウ

**⑰OPアンプによる回路設計入門**
アナログ回路の誤動作とトラブルの原因を解く

**⑲PC9801計測インターフェースのすべて**
オリジナル拡張ボードでパソコンを実践活用しよう

**⑳アナログ回路シミュレータ活用術**
ゲーム感覚の回路設計を体験しよう

**㉑ディジタル・オーディオ技術の基礎と応用**
最もポピュラーな最新技術を理解しよう

**㉒ディジタル回路ノイズ対策技術のすべて**
TTL/CMOS/ECLの活用法と誤動作/トラブルへの処方

**㉓回路デザイナのためのPLD最新活用法**
PLDのプログラミング法からPALライタの製作まで

**㉔Cによる組み込み機器用プログラミング**
16ビットCPUによるメカトロニクス入門

**㉕最新マイコン・メモリ・システム設計法**
DRAM，SRAMの動作からデュアル・ポートRAM，FIFOの活用まで

**㉖68000ソフト&ハードのすべて**
実用ライブラリの作成と便利チップ68301/68303の活用技術

**㉗ハードディスクとSCSI活用技術のすべて**
本格活用のためのハード&ソフトのすべてを詳解

**㉘最新・電源回路設計技術のすべて**
3端子レギュレータから共振型スイッチング電源まで

**㉙マイコン独習Z80完全マニュアル**
手作りの原点から実用ソフトの作成まで

**㉚ニュー・メディア時代のデータ通信技術**
赤外線，無線通信技術からLAN，光ファイバを用いた高速通信技術まで

**㉛基礎からのビデオ信号処理技術**
複合映像信号の理解からハイビジョン信号の捉え方まで

**㉜実用電子回路設計マニュアル**
アナログ回路の設計例を中心に実用回路を詳述

**㉝オプト・デバイス応用回路の設計・製作**
光素子を使いこなすための製作ドキュメント

**㉞つくるICエレクトロニクス**
機能ICを使って実用機器を作ろう〈在庫僅少〉

**㉟C言語による回路シミュレータの製作**
Quick Cでのプログラミングとフィルタ回路の解析〈在庫僅少〉

**㊱基礎からの電子回路設計ノート**
トランジスタ回路の設計からビデオ画像の編集まで

**㊲実用電子回路設計マニュアルII**
豊富な回路設計例から最適設計を学ぼう

**㊳Z80システム設計完全マニュアル**
周辺I/Oボードの設計とマイコン・システムの開発

**㊴A-Dコンバータの選び方・使い方のすべて**
アナログ信号をディジタル処理するための基礎技術

**㊵電子回路部品の活用ノウハウ**
機器の性能と信頼性を支える受動部品の使い方

**㊶実験で学ぶOPアンプのすべて**
汎用OPアンプから高性能OPアンプまで

**㊷高速ディジタル回路の測定とトラブル解析**
ハイスピード・ディジタル信号を高周波と捉らえる

**㊸Cによるマイコン制御プログラミング**
86系ペリフェラルを中心とした

**㊹フィルタの設計と使い方**
アナログ回路のキーポイントを探る

**㊺PC98シリーズのハードとソフト**
386&486マシンを使いこなす！

**㊻アナログ機能ICとその使い方**
民生用AV機器からマルチメディア分野で活躍する

**㊼高周波システム&回路設計**
通信新時代の回路技術とシステム設計

**㊽作れば解るCPU**
ロジックICで実現するZ80とキャッスル・マシン

**㊾徹底解説 Z80マイコンのすべて**
Z80 CPUの概要から周辺LSIの活用法，ICEのデバックまで

**㊿フレッシャーズのための電子工学講座**
電磁気学の基礎から電子回路の設計，製作までをやさしく解説

**(51)データ通信技術基礎講座**
RS232Cの徹底理解からローカル通信の実用技術まで

**(52)映像信号処理の基礎講座**（仮題）
映像信号の基礎からパソコンの表示技術まで
〈9月29日発売〉

CQ出版社 〒170 東京都豊島区巣鴨1-14-2 販売部☎03-5395-2141 振替 00100-7-10665

# ディスク頒布サービスのお知らせ

以下に示す各誌バックナンバのディスク頒布サービスは継続的に行っております．ご入用の方は該当雑誌で内容・価格等を確認のうえ，所定の方法でお申し込みください．なお，各申し込み書に記載の住所がスギヤマシステム宛になっている場合は，今後はCQ出版㈱デザインウェーブ宛に送ってください．

**CQ出版㈱ディスク頒布サービス係　〒170 東京都豊島区巣鴨1-14-2**

**☎ 03-5395-2113　FAX 03-5395-2107**

## トランジスタ技術

'92/6 PSpice用回路図エディタ
'92/8 マシン語モニタ・ディスク
'92/9 ロジック・アナライザ
'92/9 プリンタ・ポートに接続するPROMライタ
'92/10 ロジック・モニタを制作してROMエミュレータによるデバッグを理解しよう
'92/10 CD-Gの企画と画像再生の研究
'92/11 パソコンを利用した汎用半導体カーブ・トレーサの製作
'92/12 基礎からのC言語とパソコン制御
'93/7 ディジタル・テストパターン・ジェネレータの製作
'93/8 最強F1グランプリ・ゲーム・マシンの製作
'93/10 メタ・アセンブラ
'94/2 Z80赤外線学習リモコン
'94/3 ACパワー・コントローラ
'94/4 ロジックICチェッカ
'94/5 8チャネル・ディジタル記録計の製作
'94/8 フラッシュ・メモリ・ライタの設計と製作
'94/10 マイコン・システム設計自由自在
'94/12 トランジスタ技術データベース1994年版
'94/12 オーディオ・ビジュアル・コントローラの製作

## トランジスタ技術 SPECIAL

No.20 回路シミュレータMICRO-CAPIII
No.23 PALASM2.23D
No.33 バーコード読み取りシステムの製作とハンディ・スキャナによる画像読み取り装置の製作
No.35 C言語による回路シミュレータの製作
No.45 PC98シリーズのハードとソフト
No.49 Z80マイコンのすべて

## インターフェース

'92/1 68K用リモート・デバッガ
'92/2 バッチファイルをEXEファイルに変換する
'92/3 DOSエクステンダ用EXPファイルをEXEファイルに
'92/3 B-tree+
'92/4 プロトタイプ宣言生成ユーティリティ
'92/8 パイプ機能をより高速に使う
'92/9 パソコンで学ぶ割り込みプログラミング
'92/11 作りながら学ぶコンパイラの原理
'93/1 画素数変換の原理とCによる実現
'93/1 SCSIハードディスクユーティリティSUT
'93/2 Cで学ぶISAM処理の実際
'93/4 デバイス・ドライバをEXEファイル化する
'93/5 作りながら学ぶRS-232-Cインターフェース
'93/6 オブジェクト指向を実装するSCSIユーティリティ
'93/7 作りながら学ぶパソコン拡張インターフェース
'93/9 光ディスク・ディジタル・サーボ系設計の実際
'93/11 RS-232-C/68K系CPU/減色技法
'93/11 クライアントサーバ・プログラムを自作してみる
'93/12 μITRON/バージョンアップ検査/'93IF総目次
'94/1 Verilog，HDLシミュレータSILOSIII評価版
'94/1 科学計測におけるデータ処理技法
マニトバ版VHDL
'94/2 カオス・ジェネレータ/欧州文字変換
'94/3 Windowsプログラミング/MAKE活用技法
'94/5 計測・制御とPCインターフェース
'94/6 C++によるオブジェクト指向/RS-232-Cライン・モニタ
'94/7 DSOの設計・製作/MIDIファイルをRS-232-Cから出力する
'94/8 暗号化技法/MPEG-1 Video Audioボード/ポケベル呼び出しシステム
'94/8 DSP用ICEとアセンブラの製作
'94/10 Xbase言語はC言語を越えられるか
'94/11 エラー処理の考え方とプログラミング
'94/11 PARTHENON/CQ版用FLEXセルライブラリ
'94/12 実践Verilog-HDL-Lempel-Ziv圧縮チップの設計
'95/2 システム事例で学ぶセンシング&データ処理／PC/AT互換機3.5インチの3モード自動判別ディスク・ドライバの製作
'95/3 対話形式で学ぶCプログラミング／HD647180X簡易エミュレータの製作
'95/4 画像復元問題／KUE-CHIP2の実現
'95/5 数値演算―基礎と計測・制御での応用／WWWのためのドキュメント作成法／68K/86系リモート・デバッガを仕様変更する／KUE-CHIP2の実現／続々・COMET-ChipのRTL設計
'95/6 PCカード開発入門/Windows通信プログラムの作成と実験 HIMEMとEMM386以外をすべてUMBに逃がすユーティリティ／32ビット乗算ルーチン
'95/7 実践的HDLシステム設計入門／Windows3.1によるI/O操作の実験
'95/8 作りながら学ぶ高速／マルチポート通信／FLEXデザイン・キット/CQ版で実現する記憶力ゲームの製作／32ビット除算ルーチン／クラス・ライブラリの設計

## OPENDESIGN

No.1 SU.EXE
No.2 MAC
No.3 イーサネットとTCP/IP
No.8 電子メール

## トラ技コンピュータ

'91/1 以降の各号

# トランジスタ技術の本一覧

## トランジスタ技術SPECIAL

1～33 各¥1,540　34～45 各¥1,600　46～49 各¥1,690　50 以降各¥1,800

- 1 個別半導体素子 活用のすべて
- 3 PC9801と拡張インターフェースのすべて
- 4 C-MOS標準ロジックIC活用マニュアル
- 5 画像処理回路技術のすべて
- 6 Z80ソフト&ハードのすべて
- 8 データ通信技術のすべて
- 9 パソコン周辺機器インターフェース詳解
- 10 IBM PC&80286のすべて
- 11 フロッピ・ディスク・インターフェースのすべて
- 12 入門ハードウェア 手作り測定器のすすめ〈在庫僅少〉
- 13 シミュレータによる電子回路理論入門
- 14 技術者のためのCプログラミング入門
- 15 アナログ回路技術の基礎と応用
- 16 A-D/D-A変換回路技術のすべて
- 17 OPアンプによる回路設計入門
- 19 PC9801計測インターフェースのすべて
- 20 アナログ回路シミュレータ活用術
- 21 ディジタル・オーディオ技術の基礎と応用
- 22 ディジタル回路のノイズ対策技術のすべて
- 23 回路デザイナのためのPLD最新活用法
- 24 Cによる組み込み機器用プログラミング
- 25 最新マイコン・メモリ・システム設計法
- 26 68000ソフト&ハードのすべて
- 27 ハードディスクとSCSI活用技術のすべて
- 28 最新・電源回路設計技術のすべて
- 29 マイコン独習Z80完全マニュアル
- 30 ニューメディア時代のデータ通信技術
- 31 基礎からのビデオ信号処理技術
- 32 実用電子回路設計マニュアル
- 33 オプト・デバイス応用回路の設計・製作
- 34 つくるICエレクトロニクス
- 35 C言語による回路シミュレータの製作
- 36 基礎からの電子回路設計ノート
- 37 実用電子回路設計マニュアルII
- 38 Z80システム設計完全マニュアル
- 39 A-Dコンバータの選び方・使い方のすべて
- 40 電子回路部品の活用ノウハウ
- 41 実験で学ぶOPアンプのすべて
- 42 高速ディジタル回路の測定とトラブル解析
- 43 Cによるマイコン制御プログラミング
- 44 フィルタの設計と使い方
- 45 PC98シリーズのハードとソフト
- 46 アナログ機能ICとその使い方
- 47 高周波システム&回路設計
- 48 作れば解るCPU
- 49 徹底解説 Z80マイコンのすべて
- 50 フレッシャーズのための電子工学講座
- 51 データ通信技術基礎講座
- 52 映像信号処理の基礎講座(仮題・9月29日発売)

## トラ技ORIGINAL

各¥980（8 のみ¥1,200）

- 4 特集 実験研究CとLと回路の世界
- 7 製作実験 マイコン&メカトロニクスの誕生
- 8 実験研究 発振回路/信号発生器 完璧マスター
- 9 実験研究 アナログ・フィルタ回路の誕生

## 定本シリーズ

- 定本 発振回路の設計と応用　稲葉 保 著　¥2,800
- 定本 続トランジスタ回路の設計　鈴木雅臣 著　¥2,700
- 定本 トランジスタ回路の設計　鈴木雅臣 著　¥2,200
- 定本 OPアンプ回路の設計　岡村廸夫 著　¥2,800
- 定本 ディジタル・システムの設計　猪飼國夫／本多中二 共著　¥3,000

## 新つくるシリーズ

各¥1,500

- 1 つくるツール&測定器
- 2 つくるオーディオ&ビデオ
- 3 つくるオリジナル・グッズ

## CORE BOOKS

- 新・低周波/高周波回路設計マニュアル　鈴木雅臣 著　¥1,960
- ディジタルIC回路の設計　湯山俊夫 著　¥1,650
- 解析 ノイズ・メカニズム　岡村廸夫 著　¥1,960
- ASICの論理回路設計法　小林芳直 著　¥1,960

## ハードウェア・デザイン・シリーズ

- 電池活用ハンドブック　トランジスタ技術編集部 編　¥1,800
- ワンチップ・マイコン活用ハンドブック　トランジスタ技術編集部 編　¥1,900
- 実用電源回路設計ハンドブック　戸川治朗 著　¥1,960
- 電子回路部品活用ハンドブック　トランジスタ技術編集部 編　¥1,850
- メカトロ・センサ活用ハンドブック　トランジスタ技術編集部 編　¥1,850

## ハンドブックシリーズ

- 実用電子回路ハンドブック2〈在庫僅少〉　トランジスタ技術編集部 編　¥1,440
- 精選アナログ実用回路集　稲葉 保 著　¥2,370
- 七五調による電子回路技術定石集　稲葉 保／森口章成 共著　¥2,300

CQ出版社

下記の本の内容に関するお問い合わせはトランジスタ技術編集部(☎03-5395-2123)まで，ご購入の際には販売部(☎03-5395-2141)までご連絡ください．

## C&E TUTORIAL

- わかる電子計測技術　安孫子健一 著　¥2,700
- ハード設計ワンランク・アップ　畔津明仁 著　¥2,300
- SPICEによるOPアンプ回路の設計　岡村廸夫 著　¥2,200
- アナログICの機能回路設計入門　青木英彦 著　¥2,600
- SPICEによるトラジンスタ回路の設計　岡村廸夫 著　¥1,900
- SPICEによるシミュレータ新活用法　岡村廸夫 著　¥1,800
- SPICEによる電子回路設計入門〈在庫僅少〉　ポールW・トゥネンガ 著／松本敏之 訳　¥2,200
- パソコンで学ぶ自動制御の応用学　宮崎誠一／宮崎　仁 共著　¥2,200
- パソコンで学ぶ自動制御の実用学　宮崎誠一／宮崎　仁 共著　¥2,200
- データ伝送技術実用ノウハウのすべて　宮崎誠一 著　¥2,500
- CAEで学ぶOPアンプ回路入門　苗手英彦 著　¥2,300
- 新・オシロスコープ 測定技術のすべて　高橋　徹 編著　¥2,300
- VHDLによるハードウェア設計入門　高橋　徹 編著　¥2,300
- A VHDL Primer VHDL言語入門　Jayaram Bhasker 著／デザイン・ウェーブ企画室訳　¥3,200

## 現場技術者実戦シリーズ

- 電子回路のトラブル対策ノウハウ　稲葉　保／飯田文夫 共著　¥2,700
- 改訂 マイコン・システム設計ノウハウ　林　善雄／常田晴弘 共著　¥2,900
- 改訂 スイッチング・レギュレータ設計ノウハウ　長谷川彰 著　¥2,300
- 改訂 オプト・デバイス応用ノウハウ　谷　善平 編著　¥2,900
- 改訂 電力制御回路設計ノウハウ　在田保信／森　敏／由宇義珍 共著　¥2,000
- 改訂 高周波回路設計ノウハウ　吉田　武 著　¥2,900

## ビギナ・ノート・シリーズ

- 高周波回路の設計・製作　鈴木憲次 著　¥2,100
- ディジタル回路の設計・製作　湯山俊夫 著　¥2,200
- ディジタル回路テイクオフ指南　小林芳直 著　¥1,700
- アナログ回路の設計・製作　青木英彦 著　¥1,700
- センサ応用回路の設計・製作　松井邦彦 著　¥1,700
- 電子制御エンジンの基礎・応用　淺原　宏 著　¥2,100

## デザインウェーブ・ブックス

- 電子回路図エディタ活用マスター　倉重克己 著　¥3,000
- Verilog HDLによるトップダウン設計　E. Sternheim/R. Singh/R. Madhavan/Y. Trivedi 著　井上博史／鈴木隆 訳　¥3,900
- パソコンによるプログラマブル・ロジックの設計　トランジスタ技術編集部 編　¥3,200
- PCによるディジタル回路の設計とシミュレーション〈近日発売予定〉　トランジスタ技術編集部 編　¥3,900

## デザインウェーブ・ソフトウェア

- パソコン用プログラマブル・ロジック開発パッケージ FLEXデザイン・キット/CQ版　¥49,000
- パソコン用高位論理合成システム PARTHENON/CQ版　¥35,000
- パソコン用回路図エディタTango-SCH/CQ版　¥9,640
- パソコン用電子回路シミュレータ PSpice(CQ版)Ver.5　PC9801用　¥11,640，IBM-PC/AT用(J-3100用)　¥14,640
- パソコン用電子回路シミュレータ Micro-CapⅣ/CQ版　¥14,610
- 回路図エディタ＋電子回路シミュレータ PSpice/CQ版スーパーパック　¥14,610
- 半導体規格表総索引（FD版）　¥6,000
- 半導体規格表ディスク版 74シリーズICデータ・ライブラリ　¥9,640
- 半導体規格表ディスク版 トランジスタ・データ・ライブラリ　¥9,640
- 半導体規格表ディスク版 モノリシックOPアンプ・データ・ライブラリ　¥9,640
- 半導体規格表ディスク版 ダイオード・データ・ライブラリ　¥9,640
- 半導体規格表ディスク版 FETデータ・ライブラリ　¥9,640

## トラ技 カルチャー

- データ通信技術セミナー　宮崎誠一 著　¥1,650

## 選書シリーズ

- わかる半導体セミナー　伝田精一 著　¥1,240

## コンピュータ・ハードウェア関連

- 16ビットDOSマシン設計・製作のすべて　吉田幸作／岡崎泰三 共編　¥1,600
- Z80ファミリ・ハンドブック　額田忠之 著　¥1,960
- 8086ファミリ・ハンドブック　相沢一石 著　¥1,800

〒170 東京都豊島区巣鴨1-14-2　販売部☎03-5395-2141　振替 00100-7-10665

## ご案内の雑誌，書籍は全国有名書店で販売しております

●品切れのときは，書店にお申し込みいただければ取り寄せてもらえます．その際は，送料荷造り費がかからず，荷いたみなどの事故もございませんので，なるべく書店をご利用くださることをおすすめします（ただし，お手元に本が届くまで10日～25日ぐらいかかります）．

●ブックサービス㈱へ注文することができます．代金は商品到着時に配達料（冊数にかかわらず380円）を添えてお支払いください．1週間くらいで届きます．

**ブックサービス㈱**　〒113 東京都文京区本郷1-28-10 本郷TKビル1F　☎03-3817-0711　FAX03-3818-5969

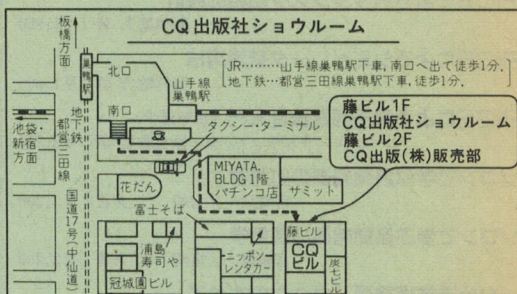
●その他に，小社販売部へお申し込みいただくこともできます．次頁をご覧のうえ，郵便振替，現金書留あるいはクレジットカードのいずれかの方法でお支払いください（郵便振替の場合，10日ぐらいかかります）．

特に明記されていない書籍等の送料荷造り費は，ご購入金額の合計により，つぎのようになります．〈平成7年8月1日より〉

| ご購入金額 | 送料荷造り費 |
|---|---|
| 2,000円未満 | 300円 |
| 2,000円以上～5,000円未満 | 350円 |
| 5,000円以上～9,000円未満 | 400円 |
| 9,000円以上 | 無料 |

**CQ出版㈱　販売部**
〒170 東京都豊島区巣鴨1-14-2　☎03-5395-2141

●CQ出版社ショウルームでは，ほとんどすべての商品を展示販売しております．ぜひ，ご利用ください（土日，祝祭日を除く10：30～17：00．ただし，8月14日～16日は夏期休暇）．

**CQ出版社ショウルーム**　☎03-5395-2143
〒170 東京都豊島区巣鴨1-14-3 藤ビル1F

●海外へ小社出版物を送る場合は，日本I.P.S.㈱へお問い合わせください．

**日本I.P.S.㈱**
〒102 東京都千代田区飯田橋3-11-6　☎03-3238-0700

CQ出版社の商品を小社販売部より購入希望される方は，つぎのいずれかの方法で代金をお支払いください．

**■郵便振替**

郵便局の振替用紙をご利用ください．通信欄にご希望の商品名を明記し，代金と送料荷作り費（前頁参照）の合計をお振り込みください．

●口座番号：00100-7-10665

**■現金書留**

この申込書に必要事項をご記入のうえ，代金と送料荷作り費（前頁参照）の合計をお送りください．

●宛先：〒170 東京都豊島区巣鴨1-14-2
CQ出版㈱ 販売部

**■クレジットカード**（ご購入金額の合計が5,000円以上の場合に限らせていただきます．）

この申込書に必要事項をご記入のうえ，FAXか封書でお送りください．

●封書宛先：〒170 東京都豊島区巣鴨1-14-2
CQ出版㈱ 販売部 クレジット係

●FAX送信先：03-5395-2106

| 送付先 |
|---|
| ご住所：□□□-□□ |
| お名前： |
| ☎（　　）　— |
| FAX（　　）　— |

お支払い方法（いずれかに○印をおつけください）　□現金書留　□クレジットカード

| | |
|---|---|
| 商品合計金額 | 円 |
| 送料荷造り費 | 円 |
| お支払い総額 | 円 |

**クレジットカードをご利用の方は，この太枠内をすべてご記入ください．**

①ご利用カード会社名（○印をおつけください）
JCB　DINERS　VISA　MASTER　AMEX

②カード会員番号

③カード有効期限　年　月

④カード名義人（カタカナでご記入ください）

⑤お支払い回数　1回払い

## ■ 現金書留・クレジットカード専用申込書 ■

| | 商品名 | 定価 | 購入希望数 |
|---|---|---|---|
| ① | 回路図エディタ＋電子回路シミュレータ<br>PSpice®/CQ版スーパーパック（PC9801用 3.5″） | 14,610円 | |
| ② | パソコン用電子回路シミュレータ（3.5″ or 5″）<br>PSpice/CQ版（Ver.5）PC9801用 | 11,640円 | 3.5″ / 5″ |
| ③ | パソコン用電子回路シミュレータ（3.5″）<br>PSpice/CQ版（Ver.5）PC/AT，J-3100用 | 14,640円 | |
| ④ | パソコン用回路図エディタ<br>Tango-SCH/CQ版（PC9801用 3.5″ or 5″） | 9,640円 | 3.5″ / 5″ |
| ⑤ | '95年版半導体規格表ディスク版<br>トランジスタ・データ・ライブラリ（PC9801用 3.5″） | 9,610円 | |
| ⑥ | '95年版半導体規格表ディスク版<br>ダイオード・データ・ライブラリ（PC9801用 3.5″） | 9,610円 | |
| ⑦ | '95年版半導体規格表ディスク版<br>FETデータ・ライブラリ（PC9801用 3.5″） | 9,610円 | |
| ⑧ | '95年版半導体規格表ディスク版<br>74シリーズICデータ・ライブラリ（PC9801用 3.5″） | 9,610円 | |
| ⑨ | '95年版半導体規格表ディスク版<br>OPアンプ・データ・ライブラリ（PC9801用 3.5″） | 9,610円 | |
| ⑩ | '95年版半導体規格表総索引（FD版）<br>（PC9801，DOS/V用 3.5″ 2DD〈720Kバイト〉） | 6,000円 | |
| ⑪ | '95年版最新半導体規格表セット<br>半導体規格表全20冊＋⑩総索引FD（3.5″） | 24,700円 | |
| ⑫ | トラ技スーパーキットDAIレシーバ<br>（「トランジスタ技術」'92年8，9月号参照） | 20,000円 | |
| ⑬ | トラ技スーパーキット実験用可変電源<br>（「トランジスタ技術」'92年10，11月号参照） | 20,000円 | |
| ⑭ | トラ技スーパーキットディジタル・マルチメータ<br>（3.5″ or 5″）（「トランジスタ技術」'93年3，4，8，9月号参照） | 20,000円 | 3.5″ / 5″ |
| ⑮ | パソコン用プログラマブル・ロジック開発パッケージ<br>FLEXデザイン・キット/CQ版<br>（Windows用（PC9801，PC/AT）3.5″） | 49,000円 | |
| ⑯ | パソコン用高位論理合成システム<br>PARTHENON/CQ版（PC9801，DOS/V用 3.5″） | 35,000円 | |
| ⑰ | パソコン用ディジタル回路エディタ&シミュレータ<br>SUSIE-CAD/CQ版（Windows用（PC/ATのみ）3.5″） | 39,000円<br>（近日発売予定） | |
| ⑱ | Verilog-HDLシミュレータ<br>Pure Speed/CQ版（Windows用（PC9801，PC/AT）3.5″） | 価格未定<br>（9月発売予定） | |

**上記の商品以外の書籍等をご希望の方は下欄をご利用ください．**

| ISBNコード | 商品名 | 定価 | 購入希望数 |
|---|---|---|---|
| 4-7898-□□□□-□ | | | |
| 4-7898-□□□□-□ | | | |
| 4-7898-□□□□-□ | | | |
| 4-7898-□□□□-□ | | | |

■②④⑭の商品につきましては，ご希望のメディアの欄に数量を明示してください．

■ご不明な点は小社販売部（☎03-5395-2141）までお問い合わせください．

# 各社データブック新刊のご案内

| メーカー | 資料名 | 内容 | 製品コード | 定価 |
|---|---|---|---|---|
| 日本モトローラ | バイポーラ・パワー・トランジスタデータブック | データシート，パッケージ情報 | 70861 | ¥742 |
| | ツェナー・ダイオードデータブック | 製品一覧,セレクタ・ガイドおよびデータブック,テクニカル・インフォメーション,アプリケーション・ノート | 70871 | ¥948 |
| | TMOSパワーMOSFETデータブック | セレクタ・ガイド，データシート，アプリケーション情報，パッケージ情報 | 70881 | ¥1,833 |
| | 圧力センサデータブック | 概要，新製品，データ・シート，品質と信頼性，アプリケーション・ノート他 | 70891 | ¥700 |
| インテル・ジャパン | Intel486プロセッサ・関連製品(英文) | i486プロセッサ，PCIセット，ペリフェラル・コンポーネンツ | 72981 | ¥2,570 |
| | Pentiumプロセッサ・関連製品(英文) | Pentiumプロセッサ，PCIセット，周辺素子 | 72991 | ¥2,261 |
| | '95 周辺コンポーネント | チップセット，PCI/O，ペリフェラル，メモリ・コントローラ，キーボード・コントローラ他 | 73001 | ¥2,467 |
| | '95 パッケージング(英文)2冊セット | 外形，寸法 | 73011 | ¥2,055 |
| 日立マイコン・システム | '95 通信用半導体 | 用途別早見表，信頼性，パッケージ，データシート，関連デバイスの紹介他 | 77391 | ¥14,729 |
| | '95 標準リニアIC | 概要，オペアンプ，ボルテージコンパレータ，シリーズレギュレータ，スイッチングレギュレータ他 | 77401 | ¥4,120 |
| | '95 高速A/D，D/Aコンバータ | A/Dコンバータ，D/Aコンバータ，アプリケーション・ノート | 77411 | ¥1,648 |
| | '95 小信号トランジスタ | 概要，バイポーラトランジスタ，電界効果形トランジスタ | 77421 | ¥9,270 |
| | '95 Bi-CMOS/低電圧CMOS標準ロジック(HD74BC/LVC) | 概要,HD74LVCシリーズ,HD74LVシリーズ,HD74BCシリーズ | 77431 | ¥3,605 |
| | '95 HDD/FDDコントローラIC | 概要，HDD用IC，FDD用IC | 77441 | ¥4,326 |
| | '95 日立モータドライバIC | 概要，モータドライバ，インターフェースドライバ | 77451 | ¥4,223 |
| NEC | '95 TV・VTR用IC | 選局・電源回路，IF，映像信号処理，偏向回路 | 77951 | ¥7,210 |
| | '95 表面実装ディスクリート・デバイス | 2SAタイプ，2SBタイプ，抵抗内蔵タイプ，μPAタイプ | 77961 | ¥4,944 |
| | '95 パワーMOS FET | Nチャネルパワー MOS FET,Pチャネルパワー MOS FET | 77971 | ¥4,326 |
| | '95 マイクロ波デバイス | ローノイズトランジスタ，パワートランジスタ | 77981 | ¥2,472 |

■上記の新刊本をご希望の方は，出版案内14をご参照のうえ，お申し込みください．

■通信販売ご利用の際，出版案内15の申込書のISBNコード欄には製品コードをご記入ください．

■上記以外のデータブック既刊本につきましては，各メーカ別に注文書を用意しておりますので，メーカーをご指定のうえ，電話またはFAXにて，下記までご連絡ください．

**CQ出版㈱ 販売部 ☎03-5395-2141 FAX03-5395-2106**

## データブック常備店 各社データブックは下記の書店でもお買い求めいただけます．

| 書店 | 所在地 | 電話 |
|---|---|---|
| 旭屋書店 札幌店 | 札幌市 | ☎011-241-3007 |
| 金港堂本店 | 仙台市 | ☎022-225-6521 |
| 大盛堂書店 | 渋谷区 | ☎03-3463-0511 |
| 書泉グランデ | 千代田区 | ☎03-3295-0011 |
| 書泉ブックタワー | 千代田区 | ☎03-5296-0051 |
| 万世書房 | 千代田区 | ☎03-3255-0605 |
| ラオックスザコンピュータ館 | 千代田区 | ☎03-5256-3111 |
| 電波堂書店 | 千代田区 | ☎03-3255-8539 |
| 三省堂書店 神田本店 | 千代田区 | ☎03-3233-3312 |
| 東京旭屋書店 池袋店 | 豊島区 | ☎03-3986-0311 |
| 有隣堂西口トーヨー地下街店 | 横浜市 | ☎045-311-6265 |
| 文教堂ブックセンター | 川崎市 | ☎044-812-0063 |
| アイブック三洋堂パソコンΣ 上前津店 | 名古屋市 | ☎052-251-8834 |
| 丸善 名古屋支店 | 名古屋市 | ☎052-261-2251 |
| アバンティブックセンター | 京都市 | ☎075-682-5031 |
| コーサカ・テクノ | 大阪市 | ☎ 06-763-5801 |
| 淳久堂書店 三宮店 | 神戸市 | ☎078-392-1001 |

郵便はがき

170

料金受取人払

豊島局承認

9053

差出有効期間
1996年3月31日まで

切手をはらずに
お出しください

（受取人）
東京都豊島区巣鴨1-14-2
CQ出版株式会社
トラ枝コンピュータ リーダーズカード 行

1995年9月号

■トラ枝コンピュータに対する感想を率直にご記入ください.

| お名前 | | 自宅電話番号 | （　） |
|---|---|---|---|

| 自宅住所 | 〒 |
|---|---|

■上記アンケートの内容　読者の声のページに掲載（可，不可）
ペンネーム（　　　　　）

●掲載広告企業からのDM（ダイレクトメール）をご希望されない方は（いいえ）に丸をつけてください.

# 資料請求カード　TC　1995年9月号

カード有効期限：1995年9月末日迄
送付先は勤務先に限ります。

当社記入欄
1 VOL 0064
2 受付

3 登録内容変更箇所No.

4 請求状況　1＝新規　2＝登録済　3＝登録内容の変更

5 会員番号 CQ　000

6 フリガナ
7 姓名　姓　名

8 生年月日　19　年　月　日

9 〒　都道府県　区市郡　区町村
10 勤務先住所

11 会社名
12 部署名

13 勤務先電話番号　市外局番（局番）電話番号

14 役職　15 所属部門　16 担当分野

17 業種　18 従業員数

19 広告資料請求欄　請求No.の欄に資料請求番号を記入し、A.B.C.のご希望の項目に○をつけてください

| 希望項目＼請求No. | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| 購入希望 | A | A | A | A | A | A | A | A | A | A |
| 説明希望 | B | B | B | B | B | B | B | B | B | B |
| 資料希望 | C | C | C | C | C | C | C | C | C | C |

| 希望項目＼請求No. | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| 購入希望 | A | A | A | A | A | A | A | A | A | A |
| 説明希望 | B | B | B | B | B | B | B | B | B | B |
| 資料希望 | C | C | C | C | C | C | C | C | C | C |

# 資料請求カード記入法

専門誌の広告は重要な技術情報です。その中で読者の皆様が入手したい情報は、資料請求カードのご利用で、居ながらにしてお手元に届きます。ぜひ有効にご活用下さい。

- ○必ず全項目にご記入下さい。
- ○姓名・勤務先・電話番号・住所は所定の様式通り1枠ごとに書き入れて下さい。
- ○会社名は株式会社ならば(株)、有限会社は(有)、財団法人は(財)と略し社名を書いて下さい。
- ○生年月日は西暦で書いて下さい。
- ○当カードは登録することができます。まず全項目にご記入の上投函下さい。当社より登録完了はがきが到着します。
- ○当社請求カードの会員番号がある方は、会員番号・姓名・生年月日の記入ですみます。また会員番号をお持ちで登録内容に変更がある方は変更箇所を上記の他に書いて下さい。
- ○役職・職種・専門分野・業種・従業員数は下記の該当する番号を記入して下さい。

**役職**
1. 一般職
2. 主任職
3. 係長職
4. 課長職
5. 部長　次長職
6. 役員職
7. 経営者
8. 教職(小、中、高校)
9. 教職(大学、専門学校)
10. 専門職(弁護士・医師・会計士等)
99. その他

**所属部門**
1. 研究・開発部門
2. 設計・技術部門
3. 生産・生産管理部門
4. 保守サービス部門
5. その他の技術部門
6. 営業部門
7. 購買・資材部門
8. 総務・経理部門
9. 人事・教育部門
10. 経営部門
99. その他

**担当分野**
1. コンピュータ(ハードウェア)技術
2. コンピュータ(ソフトウェア)技術
3. 半導体・デバイス技術
4. 電子回路技術
5. 計測・制御技術
6. 通信関連技術
7. 物性・材料
8. 機構設計
9. システムエンジニアリング
10. 検査・品質管理
11. 医師
12. 経営管理
13. 商工サービス
14. 教職
15. 学生
99. その他

**業種**
1. 建設・不動産
2. 金融・保険
3. 化学・エネルギー
4. 鉄鋼・金属
5. 機械・精密機器
6. 電子機器・部品
7. 輸送用機器・運輸
8. その他の製造
9. 商社・卸売
10. 通信・情報
11. 情報処理
12. その他のサービス
13. 政府公共機関
14. 学校・研究所・病院
15. 農林・水産業
99. その他

**従業員数**
1. 10人以下
2. 11人～50人
3. 51人～100人
4. 101人～500人
5. 501人～1000人
6. 1001人～5000人
7. 5001人～10,000人
8. 10,001人以上

# アンケートのお願い

よりよい内容の雑誌を作るために，アンケートにご協力ください．

アンケートの項目は次のページにあります．
アンケートは折り込みハガキにご記入の上
ご返送ください．

アンケートに応募していただいた方には，下記の中からご希望のモニターを選んでいただき，抽選でプレゼントいたします．

また，抽選に漏れたかたがたの中から100名様には，弊社特製のCD-ROMケースを進呈いたします．

## モニタープレゼント商品一覧

(1) 1-2-3 R2.5J (1セット)〔提供:ロータス㈱〕
(2) PowerPoint for the Macintosh Ver.4.0 (1セット)〔提供:マイクロソフト㈱〕
(3) ImageOFFICE for V1.2 (1セット)〔提供:富士通㈱〕
(4) Barrier II (5セット)〔提供:㈱ランハード〕
(5) STARFAX for Windows Ver.4.0 (1セット)〔提供:メガソフト㈱〕
(6) キッドデスク日本語版 (1セット)〔提供:㈱システムソフト〕
(7) ATLAS for Windows V2.0 (1セット)〔提供:富士通㈱〕
(8) 電辞窓 旺文社国語辞典 (1セット)〔提供:㈱メディア・ラボ〕
(9) 名刺男(メイシマン) (1セット)〔提供:パーソナルメディア㈱〕
(10) DynaFontプレミアム17書体パック Win/Mac用 (各1セット)〔提供:ダイナラブジャパン㈱〕
(11) AXP-SC02H(1セット)〔提供:㈱アドテックシステムサイエンス〕
(12) Power Window 868PCI (1セット)〔提供:カノープス㈱〕
(13) MX2-SB (2セット)〔提供:㈱アイ・オー・データ機器〕
(14) ACパワーコントローラ SPC-02シリーズ(ポン太) (2セット)〔提供:㈱システムサコム〕
(15) 日本語OCRソフト e.Typist (2セット)〔提供:メディアドライブ㈱〕
(16) ViewWave EX DVS-1632 (1セット)〔提供:緑電子㈱〕
(17) PCMCIA SCSI-2カード IFC-VS (1セット)〔提供:㈱メルコ〕
(18) ウインドウズアクセラレータVA-100(5セット)〔提供:㈱ティアックシステムクリエイト〕
(19) マルチメディアスピーカシステム(2セット)〔提供:松下電器産業㈱〕
(20) REX-5568・DATA/FAXモデムカード (1セット)〔提供:ラトックシステム㈱〕
(21) REX-5052・GP-IBカード (1セット)〔提供:ラトックシステム㈱〕
(22) REX-5054・A/Dカード (1セット)〔提供:ラトックシステム㈱〕
(23) REX-5055・DIOカード (1セット)〔提供:ラトックシステム㈱〕
(24) REX-5056・RS-232Cカード (1セット)〔提供:ラトックシステム㈱〕

**Q1 パソコン歴は何年ですか.**
1.1年未満　2.1～5年未満　3.5～10年未満　4.10～15年未満　5.15年以上

**Q2 本誌を購読して何年ですか**
1.1年未満　2.1～3年未満　3.3～5年未満　4.5年以上

**Q3 本誌の購読は.**
1.毎号必ず　2.ほぼ毎号　3.内容により　4.たまに　5.今回が始めて

**Q4 本誌を何人の方が読まれますか (回読人数).**
1.自分だけ　2.1人～2人　3.3人～4人　4.5人～6人　5.7人以上

**Q.5 本誌をどのような目的のために購読していますか** (複数回答可).
1.ソフト，周辺機器を活用するため　2.パソコン本体を活用するため　3.プログラミング技術の向上
4.市場の動向を掴むため　5.新製品の詳しい情報が知りたい　6.広告で製品情報を得るため　7.パソコン本体および周辺機器購入の参考にするため　8.企業での導入事例を知るため

**Q.6 本誌の記事を見る頻度を教えてください.**
(a) 特集記事　1.必ず見る　2.ほとんど見る　3.見る時もある　4.あまり見ない
(b) 連載の記事　1.必ず見る　2.ほとんど見る　3.見る時もある　4.あまり見ない
(c) 広告　1.必ず見る　2.ほとんど見る　3.見る時もある　4.あまり見ない
(d) 製品紹介記事　1.必ず見る　2.ほとんど見る　3.見る時もある　4.あまり見ない
(e) 新製品情報　1.必ず見る　2.ほとんど見る　3.見る時もある　4.あまり見ない
(f) コラム　1.必ず見る　2.ほとんど見る　3.見る時もある　4.あまり見ない

**Q7 日常，勤務先(仕事)やご自宅で使用しているコンピュータは.**
(複数のパソコンをお使いの場合は最も使用時間の長いものを選んでください.( )内には型番をご記入ください)
◆勤務先(仕事)　1.Mac　2.PC98　3.IBM PC/AT互換機　4.ワークステーション　5.その他　6.使っていない
◆自宅(仕事)　1.Mac　2.PC98　3.IBM PC/AT互換機　4.ワークステーション　5.その他　6.使っていない

**Q8 1日の勤務先と自宅でのパソコンのおおよその使用時間(hr)は.**
◆勤務先：( )　◆自宅：( )

**Q9 勤務先と自宅で使っているパソコンの台数は.**
◆勤務先：( )　◆自宅：( )

**Q10 勤務先と自宅で使っているパソコンはネットワークにつながっていますか.**
◆勤務先：(1.yes, 2.no )　◆自宅：( 1.yes, 2.no )

**Q11 勤務先と自宅で使っているメインのプログラム開発言語はどの会社の製品ですか.**
◆勤務先　1.ボーランド　2.マイクロソフト　3.その他( )
◆自宅　1.ボーランド　2.マイクロソフト　3.その他( )

**Q12 勤務先と自宅で使っているOSはなんですか.**
◆勤務先　1.DOS　2.DOS+Windows3.1　3.OS/2　4.WindowsNT　5.UNIX　6.その他( )
◆自宅　1.DOS　2.DOS+Windows3.1　3.OS/2　4.WindowsNT　5.UNIX　6.その他( )

**Q13 余暇にプログラムを作りますか.**
1.yes　2.no

**Q14 勤務先やご自宅で使用されているパソコンのCPUは何ですか.**
(複数のパソコンをお使いの場合は最も使用時間の長いものを選んでください)
◆勤務先　1.Pentium75MHz以上　2.486/66MHz以上Pentium66MHz以下　3.486/66MHz未満　4.386　5.その他( )
◆自宅　1.Pentium75MHz以上　2.486/66MHz以上Pentium66MHz以下　3.486/66MHz未満　4.386　5.その他( )

**Q15 今後，勤務先(仕事)や自宅(個人)で使用しているコンピュータを買い換えるとしたら何にしますか.**
◆勤務先(仕事)　1.Mac　2.PC98　3.IBM PC/AT互換機　4.その他
◆自宅(個人)　1.Mac　2.PC98　3.IBM PC/AT互換機　4.その他　( )内には型番をご記入ください.

**Q16 Q14でお答えのパソコンはどのタイプですか.**
◆勤務先(仕事)　1.デスクトップ　2.一体型　3.ノート　4.サブノート
◆個人(自宅)　1.デスクトップ　2.一体型　3.ノート　4.サブノート

**Q17 次のソフト群から使用しているものと，欲しいものをお答えください** (複数回答可).

1.ワープロ　2.表計算　3.DTP　4.CAD/CAM　5.数式処理　6.ロジック設計　7.dBASE系　8.SQL　9.エディタ　10.C言語　11.VisulaBasic　12.Delphi　13.通信　14.グラフィック　15.音楽　16.業務業種　17.プレゼンテーション　18.ゲーム　19.スケジュールソフト　20.教育・学習　21.その他(　　　　)

**Q.18 次の雑誌群のうち，ふだんお読みのものをお答えください**(複数回答可).
1.トランジスタ技術　2.インターフェース　3.CQHamRadio　4.C MAGAZINE　5.Dr. Dobbs Journal　6.The BASIC　7.Oh! PC　8.月刊ASCII　9.月刊ASCII DOS/V Issue　10.DOS/Vmagazine　11.SUPER ASCII　12.日経パソコン　13.日経バイト　14.月刊PC　15.Hello!PC　16.EYE-COM　17.Asahiパソコン　18.Paso　19.PC Fan　20.日経click　21.PC WORK　22.THE WINDOWS　23.Windows Magazine　24.WINDOWS WORLD　25.日経WinPC　26.MacUser　27.MACLIFE　28.MACPOWER　29.MacFan　30.日経MAC　31.MAC WORLD　32.TECH Mac　33.HyperLib　34.LAN TIMES　35.コンピュータ&ネットワークLAN　36.日経コミュニケーション　37.日経CG　38.UNIX USER　39.UNIX MAGAZINE　40.internet user　41.Internet Magazine　42.TECH Win　43.MdN　44.CD-ROM FAN　45.その他（　　）

**Q19 日常使っているパソコンの実装メモリは.**
1.メモリ(8M未満)　2.メモリ(8M以上16M未満)　3.メモリ(16M以上24M未満)　4.メモリ(24M以上)

**Q20 日常使っているパソコンのハードディスクの容量は**(複数ある場合は合計).
1..ハードディスク(300Mバイト以上500Mバイト未満)　2.ハードディスク(500Mバイト以上1GB未満)　3.ハードディスク(1Gバイト以上)

**Q21 日常使っているパソコンのディスプレイの大きさは**
1.ディスプレイ(15インチ以下)　2.ディスプレイ(16インチ以上19インチ未満)　3.ディスプレイ(19インチ以上)　4..白黒LCD　5.カラーLCD

**Q22 次の周辺機器のうち所有しているものと，欲しいものをお答えください**(複数回答可).
1.CD-ROMドライブ　2.MOドライブ　3.テープドライブ　4.リムーバブルHD　5.CD-ROMライタ　6.モノクロプリンタ　7.カラープリンタ　8.FAXモデム　9.フラットベットスキャナ　10.フィルムスキャナ　11.グラフィックアクセラレータ　12.サウンドボード　13.ビデオキャプチャボード　14.MIDI音源　15.MPEGボード　16.その他(　　)

**Q23 あなたは勤務先のパソコン本体および周辺機器について購入に関与していますか.**
◆パソコン本体　1.担当者　2.担当者から技術的助言を求められる　3.していない
◆周辺機器　1.担当者　2.担当者から技術的助言を求められる　3.していない
◆ソフトウェア　1.担当者　2.担当者から技術的助言を求められる　3.していない

**Q24 パソコン本体および周辺機器を購入の際，重視することを下記よりジャンル別に一つ選んでください.**
◆パソコン　1.ブランド　2.機能/性能　3.価格　4.拡張性　5.操作性　6.友人がもっている　7.広告　8.雑誌記事の評価
◆周辺機器　1.ブランド　2.機能/性能　3.価格　4.拡張性　5.操作性　6.友人がもっている　7.広告　8.雑誌記事の評価
◆ソフトウェア　1.ブランド　2.機能/性能　3.価格　4.拡張性　5.操作性　6.友人がもっている　7.広告　8.雑誌記事の評価

**Q25 パソコンの周辺機器やソフトウェアの増設/追加はどのようにされますか.**
1.自分で行う　2.パソコンショップ/保守契約会社に頼む　3.友人/知人に頼む　4.その他

**Q26 パソコンの周辺機器やソフトウェアを購入する際に，参考にするものは何ですか**(複数回答可).
1.一般誌の記事　2.専門誌の記事　3.一般誌/新聞/テレビの広告　4.専門誌の広告　5.友人／知人の意見　6.ダイレクトメール　7.Expo/ショー　8.購入先の店員の意見　9.製品カタログ/ガイド　10.パソコン通信での評判　11.インターネットのWWWサーバーの情報　12.その他(　　)

**Q27 パソコンの本体，周辺機器やソフトウェアを購入する時期は.**
1.リース更新時　2.ボーナス時　3.組織改革時　4.特定されない

**Q28 パソコンの本体，周辺機器やソフトウェアの購入はいつもはどこですか**
1.パソコン専門ショップ　2.パソコン/家電量販店　3.通信販売　4.直輸入店　店名:(　　　　　　)

**Q29 購入するショップ(通販)を決める際の要因を下記よりそれぞれ三つ選んでください.**
◆ショップ
1.価格　2.製品が展示している　3.知名度　4.店員に質問できる　5.修理/交換のサポート　6.すぐに手に入れることができる　7.雑誌に広告が掲載されている　8.友人の薦め　9.場所(利便性)
◆通販
1.信頼性　2.価格　3.商品構成　4.知名度　5.電話の応対　6.納入時間　7.サポート　8.雑誌に広告が掲載されている　9.友人の薦め

**Q30 通販利用の動機は何ですか.**
1.ショップが近くにない　2.価格が安い　3.その他

## PCMCIA SCSI-2カード AXPシリーズ

アドテックシステムサイエンス

JEIDA4.2／PCMCIA2.1 PLUG MAGICシリーズの新製品で、DOS/Vノートやサブノート用のSCSI-2カードをダイレクトにて販売を行っています。
ノートパソコンやサブノートパソコンにCD-ROMドライブ、MOドライブ、ハードディスクなどのSCSI機器を接続するときには欠かせないカードが、このPCカードです。PC-9800シリーズとDOS/Vパソコン、両方のドライバソフトを添付しているため、汎用性の高さが魅力です。接続するSCSI機器のコネクタサイズにより、AXP-SC02HまたはAXP-SC02Fがお選びいただけます。

■仕様

| | |
|---|---|
| カードサイズ | PCカードTypeIIサイズ |
| 規　　格 | ANSI規格X3.131-1986準拠SCSI-2対応 |
| 最大転送速度 | 4MB/sec(同期転送時) |
| SCSIバス | 不平衡、アクティブターミネータ内蔵 |
| ASPIマネージャ | PC-98用、DOS/V用、EPSON PC486NOTE用 |
| 専用ドライバ | HDD,CD-ROM,MO,リムーバブルディスク用添付 |
| 専用ユーティリティ | フォーマットプログラム |

DOS版オーディオCDプレーヤ等添付ケーブル
AXP-SC02H：ハーフピッチ 50ピンコネクタ(70cm)
AXP-SC02F：フルピッチ 50ピンコネクタ(70cm)

■価格 ¥19,800

問い合わせ先:
(株)アドテックシステムサイエンス
〒240・横浜市保土ケ谷区神戸町134 YBPハイテクセンター
TEL. 045-331-7575

資料請求No.7

## Power Window 868PCI・VLB

カノープス

各誌で高い評価を得たPower Windowの新型868。定評のディスプレイドライバがアクセラレータチップの能力をフルに引き出します。

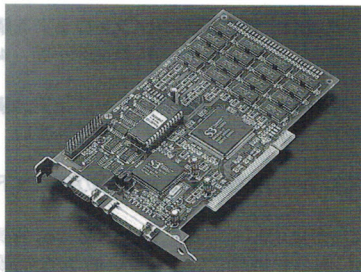
■ビデオCDとMPEG再生が可能。
スムーズな動画再生を実現するDCIにハードウェアレベルで対応。オリジナルビデオCDとプレイヤーソフト「Power Cinema」が付属していますので、その効果をすぐに確認することができます。

■鮮明でクッキリした高画質。
高精度分離型DAC回路により、Power Window 868は、高解像度・多色表示時にも、クッキリした文字と鮮明な画像表示を可能にしています。

■高い信頼性を確保。
Power Windowのディスプレイドライバは、きわめて高い信頼性を確保しています。

■高解像度の高品質表示。
800×600ドットまで可能なフルカラー表示や、使い勝手の良い1152×864ドット・65,536色をはじめとした多彩な表示モードを用意しています。また、ちらつきが少なく、目に優しいフリッカフリーモードによる表示は、1280×1024ドットまでの各解像度で対応しています。

■便利な統合ユーティリティを搭載。
解像度・色数の変更や、仮想スクリーンのコントロールなど、表示に関する設定に加え、VESA DPMS省電力機能の設定などをコントロールできるユーティリティを標準搭載しました。

■価格 ¥39,800

問い合わせ先:
カノープス(株)
〒651-22・神戸市西区室谷1-2-2 神戸ハイテクパーク内
TEL.078-992-5846

資料請求No.17

## MX2-SB サウンドボード&CD-ROMドライブセット

アイ・オー・データ機器

PC-9800シリーズ用CD-G・MKE対応の2倍速外付け型CD-ROMドライブ(CDG-MX2-L)とサウンドボード(SB16V/98)をセットしたモデルです。

■CD-G対応。

標準添付のCD-G再生ソフトで、市販のCD-G対応カラオケソフトがパソコンで楽しめます。

■PC-9800シリーズ用MKE対応。

アイ・オー・データ機器製グラフィックアクセラレータボード「GA-98NB I/C」および「SoundBlaster 16 for PC-9800シリーズ」に対応しています。

■セカンドバスTYPE II 搭載ボードに接続可能。

MKEインタフェイスボード「MK-98SB」を付ければ、セカンドバスTYPE IIボードに接続することができます。

■各種フォーマットに対応。

CD-G、Photo CD、CDマルチセッション、CD-DA、CD-ROM XA、ビデオCDの各種フォーマットに対応しています。

■CDクオリティのステレオサンプリング機能。

5～44.1KHzの範囲で選択可能な、プロ用ディジタルオーディオ機器の高音質16ビットCODECチップを採用。CDクオリティの高音質を実現しています。

■FMシンセサイザ音源搭載。

4オペレータ・20ポリフォニックステレオFMシンセサイザ(OPL3)を搭載しました。

■MIDI、ジョイスティック、外部オーディオ機器との接続が可能。

■オリジナルCD-ROMタイトルとCD-Gカラオケ体験版(絵の出るカラオケ)を添付。

●PC-9800シリーズ用

■価格:¥22,800

問い合わせ先:
(株)アイ・オー・データ機器・新製品案内窓口
〒920・石川県金沢市桜田町24街区1
TEL. 0762-60-1024

資料請求No.8

## ACパワーコントローラ(ポン太)SPC02シリーズ

システムサコム

ポン太はキーボードから、パソコン本体の電源をON/OFFできる、インテリジェントな電源タップです。Windows上では、アイコンのクリックで電源をOFFにできます。また、毎日一定時間に立ち上げるタイマー機能や、モデムの着信で自動的にONできる機能を持っていますので、ソフトウェアを作成すれば、より広い用途での利用が可能です。

■通信費のコストセーブが可能。

ポン太のタイマー機能と通信ソフトのオートパイロット機能を組み合わせると、通信費の安い夜間に自動的に立ち上げて、ダウンロードすることができます。

■モデムの着信で、Power ONが可能。

パソコンとモデムの接続ケーブル間にポン太モデム接続ケーブルを使用し、分岐した着信信号を制御ボードに接続します。ポン太はモデムから出力される着信信号を感知して、パソコンを起動します。

■モデム通信による電源投入の使用例。

ポン太(SPC02-98)と通信ソフト(インターソフト社:リモートミラー)を使ってパソコン(PC98)を遠隔操作でリモートPowerON!できます。

1.ホスト側(PC-9800)にSPC02-98を接続します。モデム接続ケーブルはRI信号用を使用し、PC-9800とモデム間のケーブルに接続します。

2.端末側のモデムからホスト側のモデムに電話をかけ、モデムから出力されるRI信号により、ホスト側の電源が投入されます。

3.電源投入後、通信ソフトを実行すると、ホスト側が無人でもファイル転送等を実行できます。

●SPC02-98:PC-9800シリーズ/EPSON PCシリーズ
SPC02-AT:DOS/V用

■価格:¥24,800

問い合わせ先:
(株)システムサコム
〒150・東京都渋谷区東1-32-12 渋谷プロパティーウエストビル12F
TEL.03-3797-0211

資料請求No.18

## 日本語OCRソフト e.Typist

メディアドライブ

マウスによるメニュー選択だけで、様々なWindowsアプリケーションに文字入力ができます。原稿をスキャナで取り込んで、簡単なメニュー選択をするだけですばやくテキストデータに変換できます。ほとんどの操作をマウスでできるため、コピーを取るような手軽さで使えます。

■すぐれた文字認識精度を実現。
書体、文字の大きさ、文字間のアキの違いにも臨機応変に対応。様々なタイプの文字を的確に認識します。インストールした時点から、精度の高い文字認識能力を発揮します。

■便利なオート機能を搭載。
原稿をセットして、操作ツールバーの「オート」を選択するだけでプレスキャンからスキャン枠設定、スキャン、認識までの一連の作業を自動的に実行します。

■日本語辞書機能を標準搭載。
約6万3千語の知識辞書「日本語ワードチェッカー」により、原稿を自動解析。いみが不自然な文字の部分を抽出します。

■簡単修正のポップアップ表示機能。
■イメージを確認しながら文字修正が可能。
■ユーザー学習機能を搭載。
■8月31日までの発売キャンペーン実施中。
■専用拡張キット「e.COPO」も好評発売中。

「e.Typist」に必要な機能だけを追加できる専用拡張キット「e.COPO」シリーズを発売。第1弾の「英文認識追加プログラム」を追加することによりバイリンガルOCRへスキルアップします。追加発売キャンペーン同時実施中。

●PC-9800シリーズ・EPSON PC・DOS/VパソコンでWindows3.1が動作する環境のパソコン用。

■価格:¥29,800（税別：キャンペーンに関するお問い合わせは下記へ）

問い合わせ先:
メディアドライブ(株)　ヒューマンメディア事業部
〒101・東京都千代田区神田小川町1-6-1 宝ビル
TEL. 03-5294-7161

資料請求No.9

## ViewWave EX DVS-1632

緑電子

FM音源、PCM音源を搭載した高音質サウンドボードです。SoundBlasterソフト対応で、ゲームも迫力のサウンドで楽しめます。また、ひと味違うホームマルチメディアとして楽しめるよう、本格的なカラオケのための特殊音響機能などもふんだんに装備しています。

■3種類の高性能音源に対応したサウンドボード。
FM音源にはSoundBlaster互換の4オペレータ・20ボイスに対応、8bit PCM音源にはSoundBlasterPRO互換の44.1khzステレオに対応し、16bit PCM音源にはWSS互換の16ビット44.1khzのCDクオリティステレオサウンド録音再生CODECチップ「CS4248」を採用し高音質が得られます。さらに、高音質なMIDI音源としてウェーブテーブル音源用チップSAM9233を採用しています。

■多彩な互換モードで、再生データが豊富。
WindowsのWAVEファイルをはじめSoundBlaster対応ソフト、MIDIソフト、WSS対応ソフトなど様々なソフトのサウンドデータが再生できます。

■本格的なカラオケのための特殊機能を装備。
マイクエコー機能や音源認識による歌の採点機能で、業務用並のカラオケが楽しめます。また、ハードウェアミキシング機能により、FM音源サウンドとマイクの音声入力をWAVEファイルとしてパソコンにCDクオリティで録音することもできます。

■パソコン背面の配線が最小限。
他の「View/Vシリーズ」との配線がパソコン内部でできるように設計されていますので、パソコンの背面の配線がスッキリします。また、9チャネルのステレオミキサを内蔵していますので、「View/Vシリーズ」のサウンドステーションボードとして機能させることができます。

●DOS／Vパソコン用

■価格:¥23,800（キャンペーン価格：¥15,800）

問い合わせ先:
緑電子(株)
〒215・川崎市麻生区下麻生559 MDテクノパーク
TEL.044-989-7623

資料請求No.19

## PCMCIA SCSI-2カード IFC-VS

メルコ

DOS/Vサブノート用のSCSI-2カードです。2048バイトのFIFOバッファを搭載し、高速なデータ転送が可能。活線挿抜機能にも対応し、電源を入れた状態でカードの抜き差しができます。PCMCIAスロットに装着すれば、これまでデスクトップパソコンでしか使えなかったCD-ROMドライブ、MOドライブ、ハードディスクなどのSCSI機器がノートパソコンで利用できます。

■FAST SCSI-2対応。

同期転送で最大10MB/S、非同期転送で最大2.4MB/Sの、高速転送が可能なFAST SCSI-2に対応しています。

■2048バイトのFIFOバッファ搭載。

効率的なデータ転送を実現するため、2048バイトのFIFOバッファを搭載しました。

■レジューム機能搭載。

電源投入時に電源OFF時のシステムの状態に復帰するレジューム機能に対応していますので、作業の再開がスムーズにできます。

■活線挿抜機能。

電源を入れたまま、PCMCIAカードを抜き差しできる活線挿抜機能を搭載。カードを抜くたびに電源ON/OFFをする必要がなく、操作性と使い勝手が向上します。

■パソコンに合わせて3タイプを用意。

DOS／Vサブノートパソコン用のIFC-VSをはじめ、NEC 98NOTE用のIFC-NS、EPSON NOTE用のIFC-ESの3タイプを用意しています。

●DOS／Vサブノート用

■価格:¥19,800

問い合わせ先:
(株)メルコ
〒457・名古屋市南区柴田本通4-15
TEL.052-619-1827

資料請求No.10

## ウインドウズアクセラレータ VA-100

ティアックシステムクリエイト

マルチメディア環境に必須の、ウインドウズ・アクセラレータボード。ノーマルモードのPC-9800シリーズをWindows適合マシンにグレードアップできます。

Windowsで、Photo CDやVideo for Windowsなどを利用する場合､ハイレゾリューション・モードが絶対条件。VA-100は、マルチスキャンディスプレイとの組み合わせで、いまご使用中のWindowsアプリケーションを高解像度・高速・多色で使用できるようにする、ウインドウズ・アクセラレータです。

■マルチレゾリューション機能。

Windows使用時、マルチスキャンディスプレイとの組み合わせで、1280×1024､1024×768､800×600､640×480､640×400の5種類から、解像度が選択できます。解像度の切換は、Windows上から実行できます。

■フルカラー1677万色。

解像度640×480または640×400で1677万色を表示できます。

■CD-ROMインタフェイス内蔵。

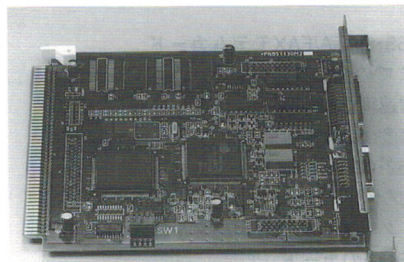
ティアックのCD-ROMユニットCD-100H専用のインタフェイスを内蔵しています。

■誰でも簡単にインストールできる、インストーラソフトを標準付属しています。

■高機能を生かす、各種サポートソフト。

- Windows3.1用ディスプレイドライバ
- 解像度／表示色数切換ユーティリティ
- ディスプレイ位置表示テストプログラム
- CD-ROMドライバインストーラ

●PC-9800およびEPSON PC用

■価格:¥24,800

問い合わせ先:
(株)ティアックシステムクリエイト
〒184・東京都小金井市本町1-18-10小金井本町ビル
TEL.0423-80-7262

資料請求No.110

## マルチメディアスピーカシステム

松下電器産業

パソコンはどんどんパワーアップし、ソフトもますますおもしろくなっています。ディスプレイも高解像度タイプが主流になったいま、クリアなサウンドへの期待が高まっています。

パナソニックのマルチメディアスピーカシステムは、豊かで厚みのある低音を再生する、独自のスーパーベースエフェクタ方式やツインロード・スーパーベース方式を採用し、なめらかでクリアな音質を実現したディスクリート・エッジスピーカなど、最新のテクノロジーを搭載。全機種アンプ内蔵で、ディスプレイへの影響が少ない防磁設計も施されています。パナソニックでは、すぐれた音質の5タイプを用意しました。

■2WAY独立アンプ方式のEAB12130(¥28,000)
ソフトの種類を問わず、ハイクオリティなサウンドを実現しました。

■3WAY方式を採用したEABMPC70(¥18,000)
高音用スピーカに高性能バランスドームツイータを採用し、クリアな高音再生を実現しています。

■広い再生帯域を持つEABMPC50(¥12,000)
サイズ以上の豊かな低音とクリアな高音が魅力です。

■超スリム設計のEAB401(¥12,800)
34mmという幅の超薄型設計で、重さも320g。コンパクトなスピーカながら、迫力は十分です。

■軽量・コンパクト設計のEABMPC30(¥7,800)
コンパクトな設計に、機能を絞り込んだシンプルで使いやすいスピーカです。すぐれた音質を確保しながら、リーズナブルな価格も実現しています。

■価格:¥7,800〜

問い合わせ先:
松下電器産業(株)
〒150・東京都千代田区外神田4-6-1 塩田トゥール秋葉原3F
TEL.03-3259-8721

資料請求No.11

## REX PCカード REX-50シリーズ

ラトックシステム

REX-50シリーズは、PCMCIA2.01/JEIDA4.1以降のTYPE-II規格に準拠した、PC-9800シリーズおよびDOS/Vパソコン対応のインタフェイスカードシリーズです。

DOS/Windows上での動作を保証し、GP-IBはハンドラを、その他のカードはサンプルプログラムのソースコード、オブジェクトコードを提供します。

また、A/Dカード用として、ディジタル信号処理プログラム/WaveLab(Windows3.1対応)をオプションで用意しています。

■REX-5568・DATA/FAXモデムカード
28,800bpsまでの全二重通信サポート。
データ通信速度:28,800bps
FAX通信速度:14,400bps

■REX-5052・GP-IBカード
GP-IBライブラリは、Visual-C/Visual-BASIC対応。
ピギーバックコネクタ付きケーブル付属。

■REX-5054・A/Dカード
分解能:12bit 8chの電圧入力A/Dコンバータ
入力レベル:0〜2.5V

■REX-5055・DIOカード
入出力:16bit(8bit単位での入出力の切換可能)
入出力レベル:TTL
セントロニクスタイプ・コネクタ付きケーブル付属

■REX-5056・RS-232Cカード
標準COMポートに割り付けられるRS-232Cカード
入出力:RS-232C非同期
D-SUB25ピン付きケーブル付属

■価格:お問い合わせ下さい

問い合わせ先:
ラトックシステム(株)
〒556 大阪市浪速区敷津東1-6-14 朝日なんばビル
TEL.06-633-2332

資料請求No.111

資料請求No.3

## Win32が世界を変える

**32bitのパワーがいままでの限界を超える。ARAGO for Windowsが**
**Xbaseの世界をWindows NT、Windows 95へ移行します。**

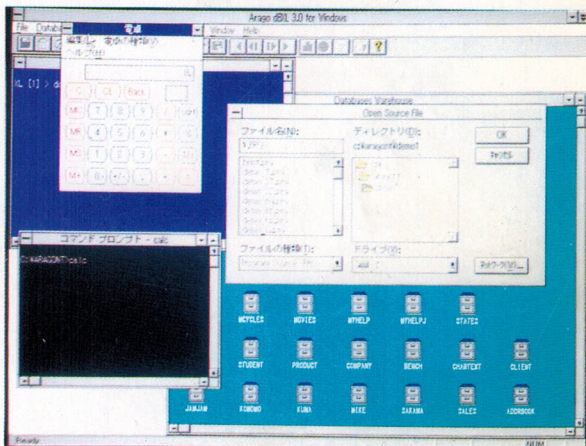
**ARAGO for WindowsでXbaseとWindowsの融合。マルチメディア、インフォメーションハイウェー、明日へのアプリケーション開発。**
**使いやすさと将来を追求しつづける、ARAGOとサザンの技術力の回答です。**
**ARAGO for WindowsはWindowsのしゃれたデザインツールではありません。もう一つのアプローチ、オフィスの明日の世界へのアプリケーション開発ツールです。**

ARAGO for Windowsで作られたアプリケーションは、ARAGO以上にWindowsらしいアプリケーションになります。メニューバーもツールバーも自由にデザイン、Windowsのユーザーインターフェイスコンポーネンツは、ほとんどサポートしています。もちろん内部32bit。Win32sでWindows 3.1、そのままWindows NT、Windows 95対応（Windows 95ユーザーインターフェイスは95年春サポート）。オフィスの要求する大規模アプリケーションに柔軟に対応。dBXL/QuickSilverシリーズは、その使いやすさと信頼性で多くの企業ユーザーに使われています。

ARAGO QuickSilver for Windowsでプログラムをコンパイル、できあがった.EXEファイルは配布自由。NT対応のビジネスアプリケーションが使いやすいXbaseで短時間で簡単に作れます。

Xbaseは現在ANSI X3J19で言語標準化を行なっています。プログラムやデータベースファイル（.DBF）は高い互換性を持っています。もちろんネットワーク対応です。

### Windows NT、Windows 95対応

ARAGO for Windowsは、Windows NT、95をターゲットに内部32bitで開発。Windows NT、95の高度に拡張されたAPIを利用したアプリケーションが作成できます。Windows NT、95では、さらに高速な運用が可能です。もちろんWin32sを利用してWindows 3.1でも使用できます。

### 画面デザインの世界を変えた「パネルペインタ」

ARAGO for DOSで画面デザインの世界を変えたパネルペインタもGUIサポート。より使いやすくWindowsのアプリケーション画面を自由に簡単に設計できます。

### ネットワーク対応

ARAGO for Windowsは標準でネットワーク対応。スタンドアロンで使っている機能やデータ資産を有効に活用できます。

**価格**（11月30日発売開始）

| | |
|---|---|
| ARAGO dBXL for Windows | ¥69,800 |
| ARAGO Professional for Windows | ¥168,000 |

**SOUTHERN PACIFIC**

**(株)サザンパシフィック**
〒220 横浜市西区南幸2-16-20 三和横浜ビル
TEL 045-314-9514　FAX 045-317-1351

※当社ではXbase標準化を推進しています。　※NIFTYにフォーラムを開設。GO FSPLで覗いて見てください。※価格、仕様等は予告なく変更になることがあります。



資料請求No.4

トラ技コンピュータ 9月号
平成3年6月22日第3種郵便物認可　平成7年9月1日(毎月1回1日発行)　第7巻　第9号　通巻64号
CQ出版社　〒170 東京都豊島区巣鴨1-14-2　☎(03)5395-2141(販売部)
定価740円(本体718円)
雑誌 06681-9
T1006681090742